



JANUARY 26, 1998  
MSSU-EIRS-ERC-98-2

# HYBRID TECHNIQUES IN COMPUTATIONAL FLUID DYNAMICS

BY:  
DR. BHARAT K. SONI  
PROFESSOR  
AEROSPACE  
ENGINEERING  
NSF ENGINEERING  
RESEARCH CENTER  
MISSISSIPPI STATE  
UNIVERSITY  
MISSISSIPPI STATE,  
MS 39762

DTIC QUALITY INSPECTED

Approved for public release  
Distribution Unlimited

DEPARTMENT OF AEROSPACE  
ENGINEERING  
MISSISSIPPI STATE UNIVERSITY  
MISSISSIPPI STATE, MS 39762

# REPORT DOCUMENTATION PAGE

*Form Approved  
OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

|   |  |   |  |  |  |   |  |
|---|--|---|--|--|--|---|--|
| <b>1. AGENCY USE ONLY (Leave blank)</b>   |  |   | <b>2. REPORT DATE</b><br>1998  |  | <b>3. REPORT TYPE AND DATES COVERED</b><br>Final 01 Dec 93 - 30 Nov 97 |   |  |
| <b>4. TITLE AND SUBTITLE</b><br><br>Hybrid Techniques in Computational Fluid Dynamics   |  |   | <b>5. FUNDING NUMBERS</b><br><br>G - F49620-94-1-0060  |  |  |   |  |
| <b>6. AUTHOR(S)</b><br><br>Dr. Bharat K. Soni   |  |   |  |  |  |   |  |
| <b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b><br><br>Engineering Research Center for Computational Field Simulation<br>P.O. Box 9627<br>Mississippi State, MS 39762   |  |   | <b>8. PERFORMING ORGANIZATION REPORT NUMBER</b><br><br>MSSU-EIRS-ERC-98-1<br>AFRL-SL-BL-TR-98-<br>0296 |  |  |   |  |
| <b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b><br><br>Dr. Leonidas Sakell, Program Manager<br>Directorate of Aerospace Sciences<br>AFOSR/NA Bolling Air Force Base<br>Washington, DC 20322-6448   |  |   | <b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b><br><br>19980410 057                              |  |  |   |  |
| <b>11. SUPPLEMENTARY NOTES</b>  |  |   |  |  |  |   |  |
| <b>12a. DISTRIBUTION/AVAILABILITY STATEMENT</b><br><br>Approved for public release,<br>distribution unlimited   |  |   | <b>12b. DISTRIBUTION CODE</b>  |  |  |   |  |
| <b>13. ABSTRACT (Maximum 200 words)</b>   |  |   |  |  |  |   |  |
| <p>In the spirit of the Air Force Office of Scientific Research's (AFOSR) initiative to promote university/industry collaborative research, a collaborative project entitled, <i>Hybrid Techniques in Computational Fluid Dynamics (CFD)</i> has been executed by the National Science Foundation Engineering Research Center for Computational Field Simulation (NSF ERC) at Mississippi State University. The industrial participants include McDonnell Douglas Company, Boeing Company, and Teledyne Brown Engineering Company. The research and development activities of this effort are focused on the advancement of methodologies to increase the efficiency, quality, and productivity of an overall CFD simulation associated with geometrically complex configurations. The progress realized in the aforementioned development is presented in this final report.</p> <p>Progress concerning efforts designed to improve the efficiency and applicability of the CFD to geometrically complex configurations utilizing hybrid techniques is presented. This progress has been brought about through the developments and enhancements in hybrid grid generation methodology and solution algorithms for associated Euler and Navier-Stokes equations. In particular, the enhancement of the two and three dimensional hybrid grid system and CFD simulation solver is presented. The development of the three dimensional hybrid grid system and solution system allowing viscous effects with complex examples is presented to demonstrate the success of these methodologies.</p> |  |   |  |  |  | <b>15. NUMBER OF PAGES</b><br>126           |  |
|   |  |   |  |  |  | <b>16. PRICE CODE</b>                       |  |
| <b>17. SECURITY CLASSIFICATION OF REPORT</b><br><br>Unclassified  |  | <b>18. SECURITY CLASSIFICATION OF THIS PAGE</b><br><br>Unclassified |  | <b>19. SECURITY CLASSIFICATION OF ABSTRACT</b><br><br>Unclassified |  | <b>20. LIMITATION OF ABSTRACT</b><br><br>UL |  |

## GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to stay *within the lines* to meet optical scanning requirements.

**Block 1. Agency Use Only (Leave blank).**

**Block 2. Report Date.** Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

**Block 3. Type of Report and Dates Covered.** State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

**Block 4. Title and Subtitle.** A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

**Block 5. Funding Numbers.** To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

|                      |                              |
|----------------------|------------------------------|
| C - Contract         | PR - Project                 |
| G - Grant            | TA - Task                    |
| PE - Program Element | WU - Work Unit Accession No. |

**Block 6. Author(s).** Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

**Block 7. Performing Organization Name(s) and Address(es).** Self-explanatory.

**Block 8. Performing Organization Report Number.** Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

**Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es).** Self-explanatory.

**Block 10. Sponsoring/Monitoring Agency Report Number. (If known)**

**Block 11. Supplementary Notes.** Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

**Block 12a. Distribution/Availability Statement.**

Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."

DOE - See authorities.

NASA - See Handbook NHB 2200.2.

NTIS - Leave blank.

**Block 12b. Distribution Code.**

DOD - Leave blank.

DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.

NASA - Leave blank.

NTIS - Leave blank.

**Block 13. Abstract.** Include a brief (*Maximum 200 words*) factual summary of the most significant information contained in the report.

**Block 14. Subject Terms.** Keywords or phrases identifying major subjects in the report.

**Block 15. Number of Pages.** Enter the total number of pages.

**Block 16. Price Code.** Enter appropriate price code (*NTIS only*).

**Blocks 17. - 19. Security Classifications.** Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

**Block 20. Limitation of Abstract.** This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

# **TECHNICAL PROJECT REPORT**

**Project** Hybrid Techniques in Computational Fluid Dynamics

**Submitted By** Dr. Bharat K. Soni, Principle Investigator  
Professor, Aerospace Engineering  
NSF Engineering Research Center for  
Computational Field Simulation  
Department of Aerospace Engineering  
Mississippi State University  
Mississippi State, MS 39762

601-325-2647  
601-325-7692 FAX

**Submitted to** Dr. Leonidas Sakell, Program Manager  
Directorate of Aerospace Sciences  
AFOSR/NA  
Bolling Air Force Base  
Washington, DC 20322-6448

**Period of Performance** December 1, 1993 – November 30, 1997

## Table of Contents

|  |           |
|--|-----------|
| <b>1. Background .....</b>   | <b>1</b>  |
| <b>2. Project Objectives .....</b>                                       | <b>2</b>  |
| <b>3. Technical Introduction: Hybrid/Generalized Grid–Flow Solvers .</b> | <b>4</b>  |
| <b>4. Hybrid Grid Generation .....</b>                                   | <b>12</b> |
| 4.1 Advancing Front Structured Grid Generator .....                      | 12        |
| 4.1.1 Elliptic Smoothing .....   | 15        |
| 4.1.2 Boundary Conditions .....  | 27        |
| 4.1.2.1 Bezier curve .....   | 27        |
| 4.1.2.2 Linear Extrapolation .....                                       | 29        |
| 4.2 Grid Trimming .....  | 29        |
| <b>5. Hybrid Flow Solver .....</b>                                       | <b>46</b> |
| 5.1 Governing Equations .....  | 46        |
| 5.2 Cell Face–Based Data Structure .....                                 | 49        |
| 5.3 Spatial Discretization .....   | 51        |
| 5.3.1 Discretization of Convective Terms .....                           | 52        |
| 5.3.2 Higher Order Schemes .....   | 56        |
| 5.3.3 Limiter Functions .....  | 59        |
| 5.3.3.1 Barth’s Limiter .....  | 61        |
| 5.3.3.2 Venkatakrishnan’s Limiter .....                                  | 61        |
| 5.4 Viscous Flux Calculations .....                                      | 63        |
| 5.5 Time Discretization .....  | 63        |
| 5.5.1 Explicit Scheme .....  | 64        |
| 5.5.2 Implicit Scheme .....  | 65        |
| 5.6 Newton Iterations .....  | 66        |
| 5.7 Approximate Analytic Jacobians .....                                 | 67        |
| 5.8 Numerical Jacobians .....  | 68        |
| 5.9 Sparse Matrix System .....   | 69        |
| 5.10 Time Step Calculation .....   | 69        |
| 5.11 Boundary Conditions .....   | 70        |
| 5.11.1 Supersonic Inflow or Outflow .....                                | 70        |
| 5.11.2 Subsonic Inflow .....   | 70        |
| 5.11.3 Subsonic Outflow .....  | 71        |
| 5.11.4 Impermeable Wall .....  | 72        |
| 5.12 Turbulence Modeling .....   | 72        |
| 5.12.1 Integral Formulation of the Turbulence Model .....                | 74        |
| 5.12.2 Numerical Procedure .....   | 75        |
| 5.12.3 Explicit Method .....   | 77        |
| 5.12.4 Implicit Method .....   | 77        |
| 5.13 Geometric Conservation Law .....                                    | 79        |
| 5.14 Grid Movement .....   | 80        |
| 5.15 Rigid Body Movement .....   | 82        |
| <b>6. Results and Discussions .....</b>                                  | <b>85</b> |
| 6.1 Validation of Two Dimensional Flows .....                            | 85        |

---

|  |            |
|--|------------|
| 6.2 Effect of Limiter Function on Convergence .....                    | 90         |
| 6.3 Effect of Approximate and Numerical Jacobians On Convergence ..... | 93         |
| 6.4 Examples of Mixed Element Type Grids .....                         | 95         |
| 6.5 Unsteady Flow Simulation .....                                     | 97         |
| 6.6 Example For Bodies in Relative Motion .....                        | 104        |
| 6.7 Validation For Three Dimension Flow Simulation .....               | 104        |
| 6.8 Preliminary Work On Parallelization .....                          | 107        |
| <b>7. University–Industry Collaboration .....</b>                      | <b>120</b> |
| <b>8. References .....</b>   | <b>121</b> |

---

## **List of Tables**

|   |    |
|---|----|
| Table 5.1 Cell Based Data Structure ..... | 49 |
| Table 5.2 Edge-Based Data Structure ..... | 50 |

## List of Figures

|  |    |
|--|----|
| Figure 3.1 Hybrid Grid for Two Element Airfoil .....   | 10 |
| Figure 3.2 Hybrid Grid for Two Element Airfoil<br>Using New Approach .....                                     | 10 |
| Figure 4.1 Grid Point Advances to Another Level in Normal Direction<br>for Two Dimension Application. ....     | 15 |
| Figure 4.2 Grid Point Advances to Another Level in Normal Direction for<br>Three Dimensional Application. .... | 15 |
| Figure 4.3 Effect of Local Elliptic Solver on Front Advancing Grid Generation.                                 | 21 |
| Figure 4.4 Effect of Local Elliptic Solver on Front Advancing Grid Generation.                                 | 22 |
| Figure 4.5 Effect of Local Elliptic Solver on Front Advancing Grid Generation.                                 | 22 |
| Figure 4.6 Rocket Grid With Zero Forcing Functions. ....   | 23 |
| Figure 4.7 Rocket Grid With Non-Zero Forcing Functions. ....   | 23 |
| Figure 4.8 C-Type Airfoil Grid With Zero Forcing Functions. ....   | 24 |
| Figure 4.9 C-Type Airfoil Grid With Non-Zero Forcing Functions. ....   | 24 |
| Figure 4.10 O-Type Airfoil Grid With Zero Forcing Functions. ....  | 25 |
| Figure 4.11 O-Type Airfoil Grid With Non-Zero Forcing Functions. ....  | 25 |
| Figure 4.12 Missile Grid With Zero Forcing Functions. ....   | 26 |
| Figure 4.13 Missile Grid With Non-Zero Forcing Functions. ....   | 26 |
| Figure 4.14 Boundary Conditions Without Smoothing. ....  | 28 |
| Figure 4.15 Boundary Conditions With Bezier Curve Smoothing. ....  | 28 |
| Figure 4.16 Six Cylinder Grids Overlap With Each Other. ....   | 30 |
| Figure 4.17 Three Element Airfoil Grids Overlap With Each Other. ....  | 30 |
| Figure 4.18 Two Element Airfoil Grids Overlap With Each Other. ....  | 31 |
| Figure 4.19 Grid Trimming Without Aspect Ratio Checking. ....  | 32 |
| Figure 4.20 Grid Trimming With Aspect Ratio Checking. ....   | 33 |
| Figure 4.21 Grid Quality Is Not Good Without Aspect Ratio Checking. ....                                       | 33 |
| Figure 4.22 Grid Quality Is Improved With Aspect Ratio Checking. ....  | 34 |
| Figure 4.23 Trimmed Grids For Two-Element Airfoil. ....  | 34 |
| Figure 4.24 Trimmed Grids For Four-Element Airfoil. ....   | 35 |
| Figure 4.25 Grid Trimming Without Boundary Point Enrichment. ....  | 35 |
| Figure 4.26 Grid Trimming With Boundary Point Enrichment. ....   | 36 |
| Figure 4.27 Grid Trimming Without Boundary Point Enrichment. ....  | 36 |
| Figure 4.28 Grid Trimming With Boundary Point Enrichment. ....   | 37 |
| Figure 4.29 Grid Trimming Without Boundary Point Enrichment. ....  | 37 |
| Figure 4.30 Grid Trimming With Boundary Point Enrichment. ....   | 38 |
| Figure 4.31 Grid Trimming Without Boundary Point Enrichment. ....  | 38 |
| Figure 4.32 Grid Trimming With Boundary Point Enrichment. ....   | 39 |
| Figure 4.33 3/4 Circle Grid With Forcing Functions. ....   | 39 |
| Figure 4.34 O-Type Two-Element Airfoil Grid. ....  | 40 |
| Figure 4.35 C-Type Two-Element Airfoil Grid. ....  | 40 |
| Figure 4.36 Five-Element Airfoil Grid. ....  | 41 |
| Figure 4.37 Three Dimensional Surface Grid. ....   | 41 |
| Figure 4.38 Three Dimensional Missile Grid. ....   | 42 |
| Figure 4.39 Three Dimensional Arbitrary Geometry Grid. ....  | 42 |
| Figure 4.40 Three Dimensional Arbitrary Geometry Grid. ....  | 43 |
| Figure 4.41 Nine Cylinder Hybrid Grid. ....  | 43 |
| Figure 4.42 Two Arbitrary Geometries Hybrid Grid. ....   | 44 |
| Figure 4.43 Two Element Airfoil Hybrid Grid. ....  | 44 |

|   |     |
|---|-----|
| Figure 4.44 Three Element Airfoil Hybrid Grid .....   | 45  |
| Figure 4.45 Four Element Airfoil Hybrid Grid .....  | 45  |
| Figure 5.1 Example of Hybrid Grid .....   | 49  |
| Figure 5.2 Areas Considered for Averaging the Conserved Variable<br>in Node and Cell Based Schemes .....  | 52  |
| Figure 5.3 Initial Conditions for the Conservation Laws .....   | 53  |
| Figure 5.4 Distribution of Flow Variables for First and<br>Second order Scheme. ....  | 56  |
| Figure 5.5 Cells Contributing to Node N1 for the Weighted Average .....   | 58  |
| Figure 5.6 Effect of Limiter Function on Linear Reconstruction .....  | 60  |
| Figure 5.7 Sparse Matrix Structure Resulting from Implicit Scheme .....   | 69  |
| Figure 5.8 Control Volume for Edge Connecting Nodes N1 and N2 .....   | 77  |
| Figure 5.9 Notations Used for Rigid Body Dynamics .....   | 83  |
| Figure 5.10 Orientation of Reference Axis and Rotated Axis .....  | 84  |
| Figure 6.11 Hybrid Grid Around NACA0012 Airfoil .....   | 85  |
| Figure 6.12 Cp Distribution Over NACA0012 Airfoil at Freestream Mach<br>Number 0.63 and Angle of Attack 2.0 Degrees .....   | 86  |
| Figure 6.13 Convergence History for Explicit and Implicit Schemes for NACA0012<br>Airfoil at Freestream Mach Number 0.63 and Angle of Attack 2.0 Degrees ..             | 87  |
| Figure 6.14 Laminar Velocity Profile Comparison With Theoretical Data .....   | 88  |
| Figure 6.15 Turbulent Velocity Profile for Reynolds Number 2.0E+06<br>Compared with Theoretical Data .....  | 89  |
| Figure 6.16 Grid Used for Turbulent Flow Over NACA0012 Airfoil .....  | 89  |
| Figure 6.17 Mach Number Contours for NACA0012 Airfoil at Freestream<br>Mach Number 0.799 and Angle of Attack 2.26 Degrees .....   | 90  |
| Figure 6.18 Pressure Coefficient Comparison over NACA0012 Airfoil at<br>Freestream Mach Number 0.799 and Angle of Attack 2.26 Degrees .....                             | 91  |
| Figure 6.19 Unstructured Grid for Scramjet Like Geometry. ....  | 91  |
| Figure 6.20 Pressure Distribution in Scramjet Engine for Freestream Mach<br>Number 3.5 .....  | 92  |
| Figure 6.21 Effect of Limiter on Convergence to Steady State Solution .....   | 93  |
| Figure 6.22 Skin Friction Distribution Over a Flat Plate at Reynolds Number<br>2x106 and Freestream Mach Number 0.5 .....   | 94  |
| Figure 6.23 Skin Friction Coefficient Distribution Over NACA0012 Airfoil<br>at Freestream Mach Number 0.799 and Angle of Attack 2.26 Degrees .....                      | 95  |
| Figure 6.24 Convergence History for Scramjet Geometry Using<br>Approximate Analytical and Numerical<br>Jacobians for CFL 5 .....  | 96  |
| Figure 6.25 Convergence History for Scramjet Geometry Using<br>Approximate Analytical and Numerical Jacobians for CFL 20 .....  | 96  |
| Figure 6.26 Convergence History for NACA0012 Airfoil at Freestream Mach<br>Number 0.63 and Angle of Attack 2.0 Degrees for CFL 50 .....                                 | 97  |
| Figure 6.27 Convergence History for Flat Plate Using Approximate Analytical<br>and Numerical Jacobians at Freestream Mach Number 0.5 and Reynolds<br>Number 30000 ..... | 98  |
| Figure 6.28 Hybrid Grid for Four Element Airfoil .....  | 98  |
| Figure 6.29 Pressure and Mach Number Distributions for a Four Element<br>Airfoil at Freestream Mach Number 0.201 and Angle of Attack 0 degree ..                        | 99  |
| Figure 6.30 Generalized Grid Containing More Than Four Nodes per Cell .....   | 101 |
| Figure 6.31 The Contour Plot of Pressure Distribution Around<br>Circular Cylinder at Freestream Mach Number 0.3 .....   | 102 |
| Figure 6.32 Grid Used for Pitching Airfoil .....  | 102 |

|  |     |
|--|-----|
| Figure 6.33 Lift History for Pitching NACA0012 Airfoil .....   | 103 |
| Figure 6.34 Pressure Distribution Around A Pitching<br>NACA0012 Airfoil .....  | 103 |
| Figure 6.35 Residual History for Pitching NACA0012 Airfoil .....   | 105 |
| Figure 6.36 Coefficients of Lift and Drag Generated During the Pitching<br>Motion of the Airfoil .....   | 106 |
| Figure 6.37 Lift History for Hybrid and Structured Grids .....   | 107 |
| Figure 6.38 Steady State Solution For Missile Geometry at Freestream<br>Mach Number 2 .....  | 108 |
| Figure 6.39 Grid, Pressure Distribution and Mach Number Distribution at<br>$t=177.52$ .....  | 109 |
| Figure 6.40 Grid, Pressure Distribution and Mach Number<br>Distribution at $t=243.5836$ .....  | 110 |
| Figure 6.41 Grid Used for Flow Simulation Over ONERA M6 Wing .....   | 111 |
| Figure 6.42 Cp Distribution Over ONERA M6 Wing at Mach Number<br>0.84 and Angle of Attack 3.06 Degrees .....   | 113 |
| Figure 6.43 Unstructured Surface Grid Used for Flow Simulations<br>Over F-15 Aircraft .....  | 113 |
| Figure 6.44 Convergence History for F-15 Aircraft Geometry .....   | 114 |
| Figure 6.45 Pressure Distribution Over F-15 Aircraft at Mach Number<br>0.9 and Angle of attack 4.84 Degrees .....  | 114 |
| Figure 6.46 Mach Number Distribution Over F-15 Aircraft at Freestream<br>Mach Number 0.9 and Angle of Attack 4.84 Degrees .....                                      | 115 |
| Figure 6.47 Cp Distribution Over the Wing Surface for F-15 Aircraft At<br>Mach Number 0.9 and Angle of Attack 4.84 Degrees .....                                     | 116 |
| Figure 6.48 Grid for a Three Element Airfoil and its Partition into Four Blocks .....  | 118 |
| Figure 6.49 Cp Distribution for Sequential and Parallel Solvers for<br>Three Element Airfoil at Freestream Mach Number 0.2 and Angle of<br>Attack 16.2 degrees ..... | 119 |
| Figure 6.50 Speed Up of Parallel Solver With the Ideal Speed Up .....  | 119 |

## **1. Background**

Mississippi State University (MSU), in collaboration with members of the National Science Foundation Engineering Research Center for Computational Field Simulation at Mississippi State University (MSU) has established a collaborative research program funded by AFOSR, McDonnell Douglas Corporation, Boeing Company and Teledyne Brown Engineering (TBE) Corporation.

The Broad-based objective of this research project is to improve understanding of fluid phenomena associated with complex three-dimensional full configurations under various physical environments utilizing hybrid techniques. In particular, the research is placed on:

- Automatic hybrid (Structured, Unstructured and mixed cells)/generalized grid generation.
- Optimal criteria for the placement of structured, unstructured and mixed cells
- Efficient data structures and data flow
- Euler and Navier-Stokes flow solvers for hybrid/generalized grids
- Training of CFD applications specialists and technology transfer

The key element of this project is the technology transfer providing one-on-one interactions between university personnel and scientists from various aerospace industries. This program is a three-year collaborative effort. This report summarizes the accomplishments and algorithms developed under this broad based program.

## 2. Project Objectives

The broad-based objective of this research project is to provide a forum through which university and industrial researchers can jointly pursue research and development pertinent to the simulation of fluid flowfields (steady and unsteady) about complex configurations over a wide range of physical conditions utilizing hybrid techniques. Upon completion, this research will significantly reduce the “response time” for addressing complex applications and will increase the quality of simulation through the improvement of numerical algorithms, surface rendering and grid generation schemes, and the manner in which graphical techniques are used to interpret computed solutions. Specifically, research focused upon advancing the current state of knowledge regarding the following items is being conducted:

- Grid Generation – Both structured and unstructured methodologies will be explored in hybrid development. In particular, the algorithms for refinement, decomposition, redistribution, and remapping algorithms developed by the principal investigator during the 1990–93 collaborative AFOSR project will be utilized in this development. Software allowing generation of both structured and unstructured blocks within a given domain decomposition will be developed. Automization, optimal criteria for structured–unstructured blocking, and efficient data structures will be explored in this hybrid multiblock development. The second objective in the grid generation will be to generate hybrid grids of arbitrary element types connected in an unstructured manner (now are referred as Generalized Grids). The approach here will be to utilize a combination of the structured–unstructured grid generation techniques to generate optimal grids for a variety of complex configurations.
- Flow Simulation Algorithm – There are numerous flow solution algorithms for structured and unstructured grids currently in widespread use around the technical community. The objective here will be to develop a finite volume algorithm to solve the Navier–Stokes equations on generalized grids. The approach will be to utilize well proven structured and unstructured techniques and adapt them for use with generalized grids. This development will also include an automatic grid adaptation algorithm for generalized grids.
- Training of CFD Application Specialists – An often overlooked aspect regarding “hardcore” CFD applications is in regard to what kind and how much training the industrial based “CFD Application Specialist” needs.

An extremely important aspect of the present research is in regard to training of both university and industrial personnel in effectively utilizing newly developed CFD related software using hybrid techniques for solving fluid flow problems associated with complex configurations. From a day-to-day, working level, people point of view, it is anticipated that this training will tend to occur naturally as a by-product of the collaborative nature of the efforts proposed herein. However, because it is felt that training issues associated with real world CFD applications are as important as the research itself, and because this embraces of the spirit of the Air Force initiative which spawned the present proposal, and objective of this effort is to take whatever steps deemed necessary to insure that substantial exchange of information and personnel between this institution and its collaborative partners take place.

With the advent and rapid development of supercomputers and high performance workstations, it is becoming possible to treat physical field problems of engineering importance on very complex regions by the numerical solution of systems of partial differential equations. Computational fluid dynamics (CFD), for instance, has progressed to the point where structured or unstructured flow solutions about complete aircraft configurations are possible.

CFD involves:

- Grid Generation – Creation of a “computational grid,” where the spatial region of interest is split up, or discretized, into many smaller regions. Two basic strategies multiblock (composite, zonal)(structured and unstructured) are widely utilized. A multitude of techniques and codes have been developed for both structured and unstructured grid generation. However, grid generation is by far the most time (labor) intensive part of the overall CFD simulation process.
- Solution Algorithms – Solving a set of partial differential equations pertinent to fluid phenomenon being analyzed. Finite difference, finite volume and finite element algorithms with both explicit and implicit formulations have all been used successfully. Codes utilizing structured multiblock grids are widely used for Navier Stokes equations and they have been validated for a variety of physical conditions and configurations. In the last few years, unstructured flow solvers have made considerable progress especially in the simulation of inviscid (Euler) flowfields. The applicability of unstructured techniques to viscous and turbulent flow simulations is being researched by various scientists/engineers.
- Numerical Flow Visualization – post-processing software designed to allow the rendering of a large amount of digital information into graphical or analog descriptions of a particular flowfield phenomena. The widely used software systems for post processing are FAST (Ref.1) and PLOT3D (Ref. 2). An unstructured capability has recently been incorporated within FAST. However, it does not support hybrid grids and further development is needed.

In the spirit of AFOSR’s initiative to promote university–industry collaborative research, a project entitled, “Numerical flow simulations around complete configuration,” was executed during 1990–93. The broad-based objective of this research project was to provide a forum through which university and industrial researchers can jointly pursue research and development pertinent to the simulation of fluid flowfields about complete flight vehicles over a wide range of flight conditions. This collaborative research effort funded by AFOSR and participating industries (McDonnell Aircraft Company, McDonnell Douglas Research and Development and Teledyne Brown Engineering Company) has been very fruitful for MSU and the industry at Mississippi State University. Significant progress in the basic development of surface grid technology, grid adaptive methodology and domain decomposition techniques was realized.

The collaborative project entitled, “Hybrid CFD techniques,” was initiated as a three year project (1993–96)

### **3. Technical Introduction: Hybrid/Generalized Grid-Flow Solvers**

Traditionally, both experimental and theoretical methods have been used to develop designs, and analysis for equipment and vehicles involving fluid flow and heat transfer. With the advent of the digital computers, a third method, the numerical approach, has emerged. Although experimentation continues to be important, especially when the flows involved are very complex, the trend is clearly toward greater reliance on computer based predictions in practical engineering applications. The development of the high-speed digital computers has had a great impact on the way in which principles from the sciences of fluid dynamics and heat transfer are applied to problems of design, analysis, and manufacture in modern engineering practices.

Many important physical processes in nature are governed by partial differential equations which are usually difficult to solve or possess no analytical solutions. With the help of digital computers, these problems can be solved numerically, and the physical phenomena can be simulated by computers. The numerical simulations of these equations are referred to as Computational Field Simulations (CFS). The CFS process involves numerical grid generation (generation of a discrete representation of surfaces or volumes for the physical domain), numerical solution algorithms (numerical solutions of governing equations of fluid dynamics), and scientific visualization (interpretation of flow characteristics of physical domains).

During the past decade, computational simulation of fluid flow over complex configurations has progressed significantly, and numerous notable successes have been reported in the literature (Lohner [1], Marcum and Weatherill [2], Mavriplis and Jameson [3], Mavriplis and Venkatakrishnan [4], Morgan et al. [5], Venkatakrishnan and Mavriplis [6], Whitaker et al. [7], Whitfield et al.[8]) . However, the generation of a high quality mesh for such problems has often been reported as a pacing item. Hence, much effort has been expended to speedup

---

this portion of the simulation process, resulting in several approaches to grid generation. The generation of high quality meshes for such complex configuration problems is the vital part of computation field simulations because the meshes are required to accurately model the physical domains and also the mesh concentration is desired in the regions of the domains where flow features develop. The quality of the mesh for a practical problem can greatly affect the accuracy and efficiency of the solutions of the problem. The solution of the governing equations can be greatly simplified and the computational efficiency and accuracy can be improved by well-constructed set of points. Usually, the grid generation is the most labor and time consuming part among the whole computational field simulation processes. Two of the most common approaches for grid generation are based on structured multi-block (Dannenhoffer [9], Shih and Soni [10], Thompson [11]) and unstructured procedures (Kallinderis [12], Khawaja [13], Lohner and Parikh [14], Marcum and Weatherill [15], Mavriplis [16], Mavriplis [17], Weatherill [18]).

In the case of structured grids, the physical domain of interest is decomposed into a number of quadrilaterals for two dimensions and hexahedrons for three dimensions. These regions, called cells, are numbered in such a way that the neighbors of a particular cell can be determined in a trivial manner. The structured grid generation technology has two major approaches: algebraic interpolation approach which generates grids directly by interpolation with the features of economical, fast, and precise spacing control, and partial differential equations approach which generates grids indirectly by solving a set of partial differential equations. However the elliptic type Laplacian partial differential equations approach can avoid grid line crossing and the elliptic type Poisson partial differential equations can achieve grid line orthogonality with spacing controls.

In the case of unstructured grids, the flow domain is divided into triangular cells for two dimensional cases and tetrahedrons for three dimensional cases. The cell numbering does

not follow any particular pattern. Therefore, the cells that neighbor a given cell are not directly deducible. This necessitates the creation of an explicit connectivity table that contains this information and results in an overhead for storing the same. However, unstructured grids offer greater geometric flexibilities. The generalized data structure of an unstructured grid is very useful in the refinement or de-refinement of the grid since the data structure has to be changed only locally. This helps in adaptation of the grid to the flow features by adding more points where the gradients of the flow properties are significant ( Baum, et al. [19], Marcum and Weatherill [20], Mavriplis [17]) and removing points from the regions where there are no flow features of interest. Therefore, grid adaptation is easier in the case of unstructured grids as compared to moving or adding grid points to a structured grid.

A class of problems of interest to the Computational Fluid Dynamics (CFD) community is the flow simulation over dynamically moving bodies. Chimera or overset grids are the primary choice for flow simulation of this type of problems when using structured grid methodology. In this approach, complex configurations are decomposed into simpler geometric entities and structured grids are generated around them. The governing equations are solved on these individual grids and the solution variables are transferred appropriately between the grids for the next time step. The disadvantage of this approach is that the interpolation between the grids may not always be conservative. If there are discontinuities in the flow variables near the overlapping grid regions, then there may be spurious oscillations near the grid interfaces.

Flow solvers based on structured grids tend to be computationally more efficient than those based on unstructured grids. High aspect ratio cells that are necessary for the resolution of the viscous layers can be easily generated for structured grids. The state of the art for structured flow solvers, (Cooper and Sirbough [21], Whitfield et al.[8], Thomas, et al.[22]), can handle this type of high aspect ratio cells. For complex configurations, the physical domain

---

has to be decomposed into different sub-domains (blocks) and the grid has to be generated separately for individual blocks. In many cases, grid lines do not exhibit continuity across the block interfaces. Even with this relaxation of the continuity of the grid lines at the block interface, this grid generation process is time consuming. Flow solvers supporting non-contiguous interfaces require specialized interpolation procedures which, furthermore, may not ensure conservation of the flow variables at the block interface.

Flow solvers based on an unstructured mesh require more CPU time per grid point. Furthermore, the number of grid points necessary to resolve the boundary layers using nearly equilateral triangles are enormous, resulting in significantly higher CPU and memory requirements. Recently, success has been reported in generating high aspect ratio unstructured grids for viscous simulations using an advancing normal point placement strategy (Marcum [23], Pirzadeh [24]). In this case, the triangles inside the boundary layer are very skewed. This forces one to avoid the use of a cell-centered finite volume approach, as the truncation error is inversely proportional to the sine of the minimum angle of the triangles that form the mesh. Consequently, the truncation error will be very high in the cells that are in the boundary layer region. For the node centered schemes, the diagonal edges do not contribute to the effective flux balance. This results in unnecessary computation of the fluxes crossing the diagonal edges.

Hybrid or generalized element grid generation and solution techniques have been developed with the objective of combining the attractive features of both structured and unstructured techniques. The Table 1 compares the advantages and disadvantages of different grid generation approaches and there is a clear advantage for the hybrid approach.

|                         | <b>Structured</b> | <b>Unstructured</b> | <b>Chimera</b> | <b>Hybrid</b> |
|-------------------------|-------------------|---------------------|----------------|---------------|
| Geometric Flexibility   | –                 | +                   | +              | +             |
| Grid Adaptation         | –                 | +                   | –              | +             |
| High Aspect Ratio Cells | +                 | –                   | +              | +             |
| Moving Grids            | –                 | +                   | +              | +             |
| Interpolation           | –                 | +                   | –              | +             |
| Memory                  | +                 | –                   | +              | ?             |
| CPU                     | +                 | –                   | +              | ?             |

**Table 1 Relative Advantages of Different Grid Generation Approaches**

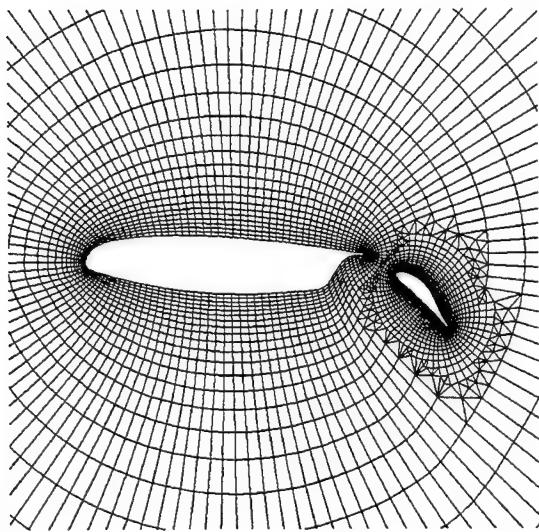
Nakahashi et al. [25] used a zonal method for the hybrid grid generation. In their approach, body fitted structured grid is kept in the boundary layer and finite difference scheme is used to solve the Reynolds averaged Navier–Stokes equation in the structured portion of the grid. The rest of the domain is filled with unstructured grid and the Euler equations are solved in that part. The interface between the structured and unstructured parts are treated as explicit boundary condition. Lohner [26] used a combination of semi-structured and unstructured grid for getting high aspect ratio cells in the boundary layer. The semi-structured grid in the boundary layer is generated using the surface normals. In this approach prisms are generated in the boundary layer and are trimmed to avoid the grid crossing. These prisms are subdivided into tetrahedra to only work with tetrahedra in the domain. The main disadvantage of the method is that the presence of highly skewed tetrahedra in the boundary layer. Kallinderis et al. [12], and Sharov and Nakahashi [27] used a semi-structured prisms in the viscous regions and tetrahedra in the rest of the domain. In both these cases, they used the marching direction as the surface normals to generate the prisms. Sharov and Nakahashi [27] used a Delaunay triangulation for the tetrahedra generation while Kallinderis et al. [12] used an advancing front method.

---

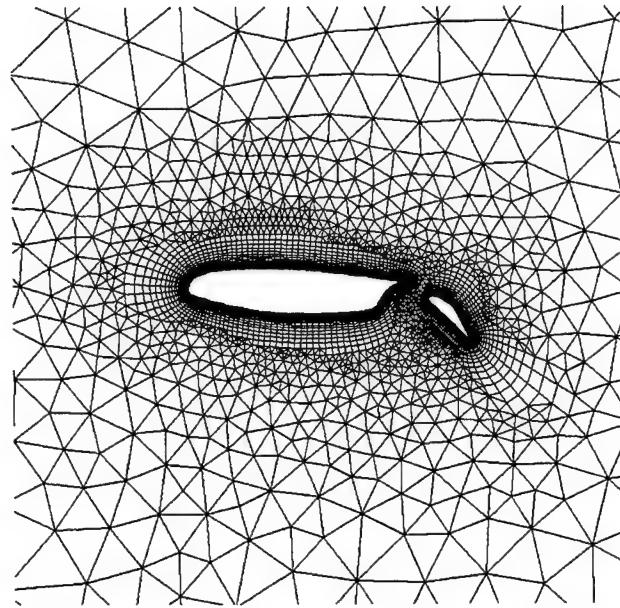
In the approach put forth by Weatherill [28] and Kao and Liou [29], most of the domain is filled with structured elements and the different components of the structured grid are connected using unstructured grids. This involves the decomposition of the complex geometries into a number of simple geometric entities followed by the generation of structured grids around them. One of the structured grids is termed the main grid and it contains the remaining grids called component grids. The main and component grids are overlaid and a hole is cut in the main grid where the structured – grid component has to be placed. The gap between the main and the component grids is filled with an unstructured grid. An example of a hybrid grid generated using this approach is shown in Figure 3.1. As can be seen from the figure, the quality of the grid at the transition between the structured and unstructured grids is not satisfactory with the area ratio of the cells varying abruptly across the interface. This leads to a higher truncation error during the discretization of the governing equations.

In the present approach, structured grids are used only near solid bodies and the rest of the domain is filled with unstructured grids. The structured grids are generated using an advancing layer type method and the quality of the cells at the interfaces is assured by checking the aspect ratios of the cell (Huang [30]). The hybrid grids are generated using a combination of structured grid generator based on an advancing layer method and an unstructured grid generator based on Delaunay triangulation. An example of the hybrid grid around a two element airfoil using the present grid generation approach is given in Figure 3.2. One can clearly observe the improvement in grid quality in the transition region between Figure 3.1 and Figure 3.2.

The development of a flow solver for generalized grids is a challenging problem, since it has to handle cells with arbitrary number of sides. The existing structured and unstructured grid algorithms are combined with the flow solver to develop a flow simulation system for hybrid grids.



**Figure 3.1 Hybrid Grid for Two Element Airfoil**



**Figure 3.2 Hybrid Grid for Two Element Airfoil  
Using New Approach**

In other works related to the flow solvers for hybrid grids, the structured and unstructured grids are treated as two different blocks (Mathur [31], Tsug et al. [32], Soetrisno et al. [33]) and explicit boundary conditions are used to transfer information between them. This

---

introduces a time lag between the interface of the structured and the unstructured grids resulting in a possible convergence degradation. In the present work, the structured and unstructured grids are treated as a single block for the purpose of flow computations. This is achieved through a generalized data structure. Another existing hybrid flow solver is due to Parthasarathy et al.[34]. In their approach, they use Lax–Wendroff temporal discretization.

A cell centered finite volume upwind scheme based on Roe’s approximate Riemann solver is used for the inviscid flux evaluation. The turbulence viscosity is estimated using the Spalart–Allmaras one equation turbulence model (Spalart and Allmaras [35]) while Sutherland’s law is used for the molecular viscosity. Both explicit and implicit schemes are implemented and validated with experimental data. The convergence of the implicit scheme using approximate analytical Jacobians and numerical Jacobians is studied. The approximate analytical Jacobians are calculated assuming the Roe averaged matrix as constant and differentiating the numerical flux crossing the cell face. But the numerical Jacobians are evaluated by perturbing the the conserved variables by a small amount and estimating the change in the numerical flux that cross the cell faces. For dynamically moving bodies, the grid movement is computed through a spring analogy and the trajectories of these bodies are determined using the laws of classical mechanics.

## 4. Hybrid Grid Generation

The overall hybrid grid generation procedure can be divided into four basic steps. In the first step the complex geometries are decomposed into number of simples geometric entities. In the next step structured grid are generated around each of these geometric entities using an advancing layer type method. In the third step, the structured grid around all geometric entities are kept together and the overlapping regions are trimmed by checking the aspect ration. In the final step the void between the trimmed structured grids are filled using unstructured grids. These steps are explained in detail in the following sections.

### 4.1 Advancing Front Structured Grid Generator

An advancing front structured grid generator is developed for generating a grid which originally marches the front out from the boundary with appropriate packing. This advancing front structured grid generator can be either used alone for generating complex geometry grid efficiently or coupled in the hybrid grid generation system.

Two-dimensional advancing front grid equations can be written as Soni[36]:

$$\begin{aligned} r_\xi \cdot r_\eta &= 0 \\ r_\eta \cdot r_\eta &= d^2 \end{aligned} \tag{4.1}$$

or  $|r_\xi \times r_\eta| = A$

or an alternative form as:

$$\begin{aligned} x_\xi x_\eta + y_\xi y_\eta &= 0 \\ x_\eta^2 + y_\eta^2 &= d^2 \end{aligned} \tag{4.2}$$

or  $x_\xi y_\eta - y_\xi x_\eta = A$

where

|               |                           |
|---------------|---------------------------|
| $r = (x, y)$  | Physical space            |
| $(\xi, \eta)$ | Computational space       |
| $d :$         | Distance for marching out |
| $A:$          | Area for marching out     |

From this system of equations, we have already known the object boundary points distributions and the distance for marching out in the current level, that is,  $r_\xi$  and  $d$  are known values. We can solve the equations for  $r_\eta$  to locate the points for another level. By solving this system of equations, the properties of orthogonality and point distributions in  $\eta$  direction are obtained.

Three-dimensional front advancing grid equations can be similarly written as Soni [36]:

$$\begin{aligned} r_\xi \cdot r_\zeta &= 0 \\ r_\eta \cdot r_\zeta &= 0 \\ r_\zeta \cdot r_\xi &= d^2 \end{aligned} \tag{4.3}$$

or as:

$$\begin{aligned} x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta &= 0 \\ x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta &= 0 \\ x_\zeta^2 + y_\zeta^2 + z_\zeta^2 &= d^2 \end{aligned} \tag{4.4}$$

where

|                      |                           |
|----------------------|---------------------------|
| $r = (x, y, z)$      | Physical space            |
| $(\xi, \eta, \zeta)$ | Computational space       |
| $d :$                | Distance for marching out |

The equations described above for two-dimensional and three-dimensional advancing front structured grid generator is nothing but a set of hyperbolic partial differential equations. An alternative three-dimensional system can be used to compute the high level points more efficiently.

$$r_\xi \times r_\eta = r_\zeta \quad (4.5)$$

$$r_\zeta \cdot r_\zeta = d^2$$

or  $r_\zeta \cdot (r_\xi \times r_\eta) = V$

$$y_\xi z_\eta - y_\eta z_\xi = x_\zeta$$

$$z_\xi x_\eta - z_\eta x_\xi = y_\zeta$$

$$x_\xi y_\eta - x_\eta y_\xi = z_\zeta \quad (4.6)$$

$$x_\zeta^2 + y_\zeta^2 + z_\zeta^2 = d^2$$

or  $x_\xi y_\eta z_\zeta + x_\xi y_\xi z_\eta + x_\eta y_\xi z_\zeta$   
 $- x_\xi y_\eta z_\xi - x_\eta y_\xi z_\xi - x_\xi y_\zeta z_\eta = V$

where

$r = (x, y, z)$  Physical space

$(\xi, \eta, \zeta)$  Computational space

$d$ : Distance for marching out

$V$ : Volume for marching out

Figure 4.1 and Figure 4.2 illustrate how the grid line or grid surface marches out from base line or surface to another level.

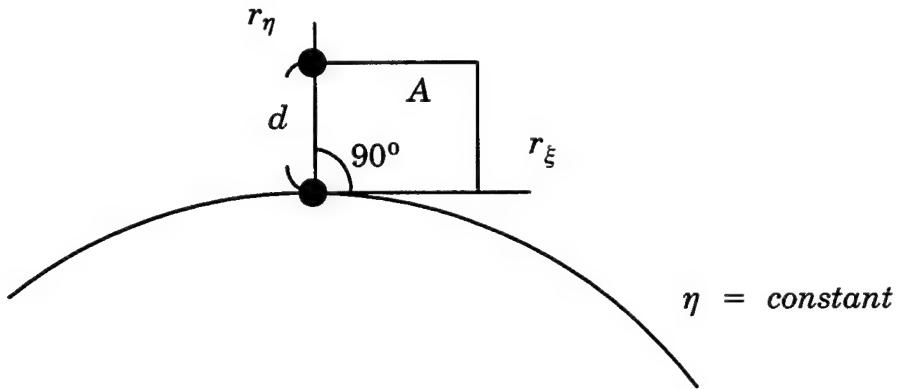


Figure 4.1 Grid Point Advances to Another Level in Normal Direction for Two Dimension Application.

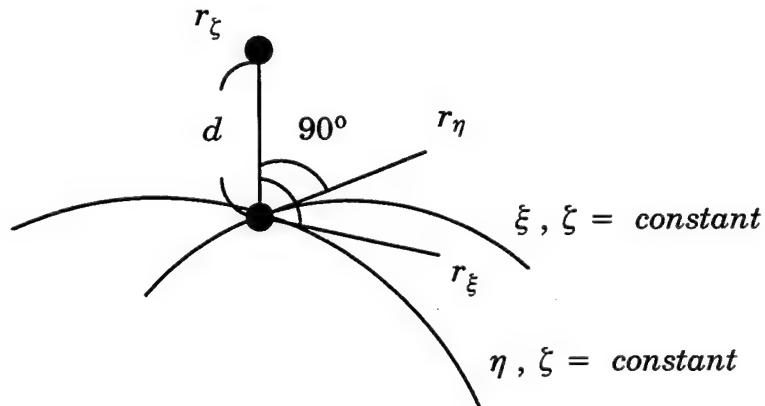


Figure 4.2 Grid Point Advances to Another Level in Normal Direction for Three Dimensional Application.

#### 4.1.1 Elliptic Smoothing

The advancing front structured grid generation has properties of orthogonality where the grid points march along the normal direction of the previous level of boundary or surface, and precise packing distribution which are supplied by known distribution data, but does not have guaranty of grid crossing resistant. For a smooth convex boundary, there is no problem for generating a perfect grid by utilizing front advancing structured grid generator. But for a

concave boundary area, it would cause grid line crossing after several steps of grid line marching. To resolve this problem, an elliptic PDE grid smoother is employed to smooth the grid at every new level to prevent grid line crossing. However, elliptic grid smoother which prevents grid line crossing diffuses the grid so that all point concentrations, and line orthogonality are lost while solving the elliptic system iteratively. Forcing control functions are introduced to accomplish field orthogonality and spacing control by elliptic PDE grid system. Therefore, instead of solving the Laplace equations, we solve the Poisson equations to smooth existed grids.

Equations of elliptic system in two-dimensional and in three-dimensional domains can be written as:

$$\begin{aligned}\xi_{xx} + \xi_{yy} &= P(\xi, \eta) \\ \eta_{xx} + \eta_{yy} &= Q(\xi, \eta)\end{aligned}\tag{4.7}$$

*(x, y) Physical Space* *(ξ, η) Computational Space*

and

$$\begin{aligned}\xi_{xx} + \xi_{yy} + \xi_{zz} &= P(\xi, \eta, \zeta) \\ \eta_{xx} + \eta_{yy} + \eta_{zz} &= Q(\xi, \eta, \zeta) \\ \zeta_{xx} + \zeta_{yy} + \zeta_{zz} &= R(\xi, \eta, \zeta)\end{aligned}\tag{4.8}$$

*(x, y, z) Physical Space* *(ξ, η, ζ) Computational Space*

When  $P$ ,  $Q$ , and  $R$  are all zero, the above equations will reduce to the Laplace equations which will guarantee the removal of grid line crossing, grid line orthogonality and a smooth grid in the field. But the distributions and concentrations of the grid will be disturbed. To preserve the grid distribution, control functions are introduced where  $P$ ,  $Q$ , and  $R$  are not zero.

The Poisson equations can be rewritten as a form of  $P$  and  $Q$  (Soni [36]),

$$g_{22} r_{\xi\xi} - 2g_{12} r_{\xi\eta} + g_{11} r_{\eta\eta} = -g(r_\xi P + r_\eta Q) \quad (4.9)$$

or a form of  $\phi$  and  $\psi$ :

$$g_{22}(r_{\xi\xi} - \phi r_\xi) - 2g_{12} r_{\xi\eta} + g_{11}(r_{\eta\eta} - \psi r_\eta) = 0 \quad (4.10)$$

or an alternative form:

$$\begin{aligned} & g_{22}(r_{\xi\xi} \cdot r_\xi - \phi r_\xi \cdot r_\xi) - 2g_{12} r_{\xi\eta} \cdot r_\xi \\ & + g_{11}(r_{\eta\eta} \cdot r_\xi - \psi r_\eta \cdot r_\xi) = 0 \end{aligned} \quad (4.11)$$

$$\begin{aligned} & g_{22}(r_{\xi\xi} \cdot r_\eta - \phi r_\xi \cdot r_\eta) - 2g_{12} r_{\xi\eta} \cdot r_\eta \\ & + g_{11}(r_{\eta\eta} \cdot r_\eta - \psi r_\eta \cdot r_\eta) = 0 \end{aligned} \quad (4.12)$$

where

$$\begin{aligned} g_{11} &= r_\xi \cdot r_\xi, \quad g_{12} = r_\xi \cdot r_\eta \quad \text{and} \quad g_{22} = r_\eta \cdot r_\eta \\ r &= (x, y) \\ P &= -\frac{g_{22}}{g} \phi \quad \text{and} \quad Q = -\frac{g_{11}}{g} \psi \\ g &= ||r_\xi \times r_\eta||^2 \end{aligned}$$

Thomas–Middlecuff proposed equation (4.13) for the control functions  $\phi$  and  $\psi$  by assuming that  $g_{12}$  equals to zero for orthogonality and one dimensional approach for each direction. Soni proposed equations (4.14) for the control functions  $\phi$  and  $\psi$  which take contributions from arc length factor for each direction. Equations (4.13) and (4.14) reveal the formula of the control functions  $\phi$  and  $\psi$  respectively from Thomas–Middlecuff and Soni [37].

$$\phi = \frac{r_{\xi\xi}}{r_\xi} \quad \text{and} \quad \psi = \frac{r_{\eta\eta}}{r_\eta} \quad (4.13)$$

$$\phi = \frac{r_{\xi\xi} \cdot r_\xi}{r_\xi \cdot r_\xi} \quad \text{and} \quad \psi = \frac{r_{\eta\eta} \cdot r_\eta}{r_\eta \cdot r_\eta} \quad (4.14)$$

Soni proposed another form of control functions which can be derived from equations (4.15) and (4.16) as following (Soni [38])

$$g_{22} \left( \frac{(g_{11})_\xi}{2} - \phi g_{11} \right) - 2g_{12} \frac{(g_{11})_\eta}{2} + g_{11} \left( (g_{12})_\eta - \frac{(g_{22})_\xi}{2} - \psi g_{12} \right) = 0 \quad (4.15)$$

$$g_{22} \left( (g_{12})_\xi - \frac{(g_{11})_\eta}{2} - \phi g_{12} \right) - \frac{(g_{22})_\xi}{2} + g_{11} \left( \frac{(g_{22})_\eta}{2} - \psi (g_{22}) \right) = 0 \quad (4.16)$$

These equations can take the contributions both from the arc length distribution factor and curvature factor into the control functions with same assumptions for orthogonality and one dimensional approach, which are :

$$\begin{aligned} \phi &= \frac{r_{\xi\xi} \cdot r_\xi}{r_\xi \cdot r_\xi} + \frac{r_{\eta\eta} \cdot r_\xi}{r_\eta \cdot r_\xi} \\ \psi &= \frac{r_{\eta\eta} \cdot r_\eta}{r_\eta \cdot r_\eta} + \frac{r_{\xi\xi} \cdot r_\eta}{r_\xi \cdot r_\xi} \end{aligned} \quad (4.17)$$

Similarly, three dimensional  $\phi$  and  $\psi$  form of Poisson equations with assumption of orthogonality for grid lines inside can be written as:

$$\begin{aligned} g_{22}g_{33}(r_{\xi\xi} - \phi r_\xi) + g_{11}g_{33}(r_{\eta\eta} - \psi r_\eta) \\ + g_{11}g_{22}(r_{\zeta\zeta} - \theta r_\zeta) = 0 \end{aligned} \quad (4.18)$$

Control functions derived from equation (4.18) are

$$\begin{aligned}
 \phi &= \frac{r_{\xi\xi} \cdot r_\xi}{r_\xi \cdot r_\xi} + \frac{r_{\eta\eta} \cdot r_\xi}{r_\eta \cdot r_\eta} + \frac{r_{\zeta\zeta} \cdot r_\xi}{r_\zeta \cdot r_\zeta} \\
 \psi &= \frac{r_{\xi\xi} \cdot r_\eta}{r_\xi \cdot r_\xi} + \frac{r_{\eta\eta} \cdot r_\eta}{r_\eta \cdot r_\eta} + \frac{r_{\zeta\zeta} \cdot r_\eta}{r_\zeta \cdot r_\zeta} \\
 \theta &= \frac{r_{\xi\xi} \cdot r_\zeta}{r_\xi \cdot r_\xi} + \frac{r_{\eta\eta} \cdot r_\zeta}{r_\eta \cdot r_\eta} + \frac{r_{\zeta\zeta} \cdot r_\zeta}{r_\zeta \cdot r_\zeta}
 \end{aligned} \tag{4.19}$$

One of the characteristics of advancing front structured grid generator is that the grid may not look good except for the first grid line because all other grid lines are derived from the previous line with normal vectors computed in the previous line. So, when we compute the forcing control functions from the whole grid, it may result bad control functions if the grid lines are crossing in the higher portion of the grid. We can rewrite equations (4.17) and (4.19) to prevent the problem by computing the control functions from the first grid line in  $\xi$  direction and from distribution in  $\eta$  direction. Equations (4.20) and (4.21) are presented for two-dimensional and three-dimensional control functions respectively. The physical meanings of terms  $g_{11}$ ,  $g_{22}$ , and  $g_{33}$  are distances of neighboring points in  $\xi$ ,  $\eta$ , and  $\zeta$  directions respectively. For hyperbolic grid generation, the grid lines marching out from the original front in  $\eta$  direction in two dimensional generation, and in  $\zeta$  direction in three dimensional generation. We have already known  $g_{22}$ , and  $g_{33}$  from the user supplied marching distribution data. Thus, we can get precise distribution control by solving Poisson equations with forcing control functions supplied in equations (4.20) and (4.21) for two-dimensional and three-dimensional grid respectively.

$$\begin{aligned}\phi &= \frac{1}{2} \left( \frac{(g_{11})_\xi}{g_{11}} - \frac{(g_{22})_\xi}{g_{22}} \right) \\ \psi &= \frac{1}{2} \left( \frac{(g_{22})_\eta}{g_{22}} - \frac{(g_{11})_\eta}{g_{11}} \right)\end{aligned}\tag{4.20}$$

$$\begin{aligned}\phi &= \frac{1}{2} \left( \frac{(g_{11})_\xi}{g_{11}} - \frac{(g_{22})_\xi}{g_{22}} - \frac{(g_{33})_\xi}{g_{33}} \right) \\ \psi &= \frac{1}{2} \left( \frac{(g_{22})_\eta}{g_{22}} - \frac{(g_{33})_\eta}{g_{33}} - \frac{(g_{11})_\eta}{g_{11}} \right) \\ \theta &= \frac{1}{2} \left( \frac{(g_{33})_\zeta}{g_{33}} - \frac{(g_{11})_\zeta}{g_{11}} - \frac{(g_{22})_\zeta}{g_{22}} \right)\end{aligned}\tag{4.21}$$

For a free boundary condition type grid, the marching distances at every grid points are the same according to the global distribution data. That means  $(g_{22})_\xi$  of equation (4.20) and  $(g_{33})_\xi$ , and  $(g_{33})_\eta$  of equation (4.21) are zero. Both the equations could be rewritten as equations (4.22) and (4.23).

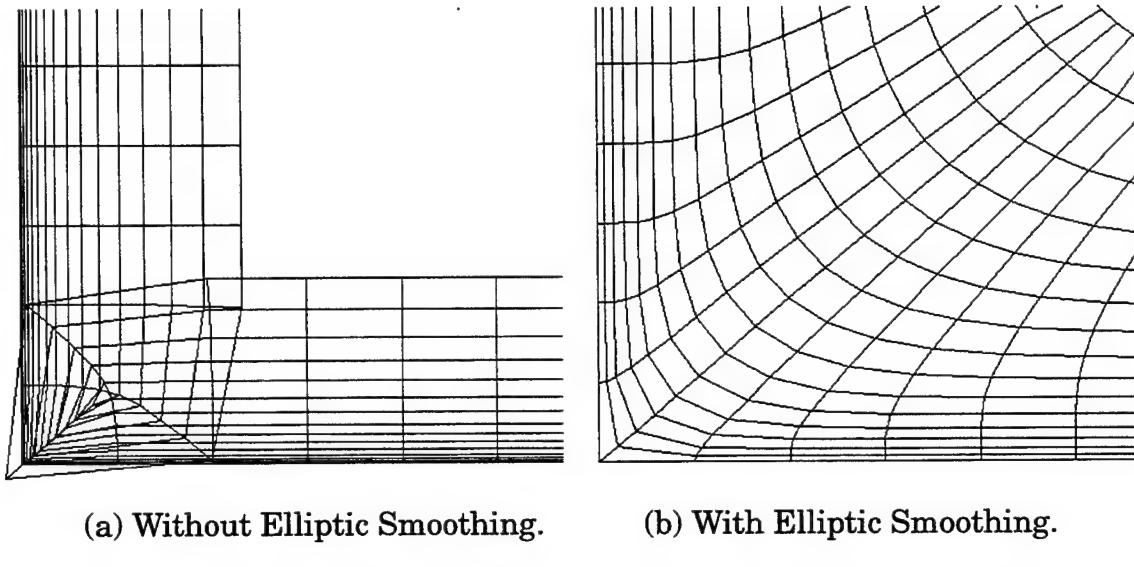
$$\begin{aligned}\phi &= \frac{1}{2} \frac{(g_{11})_\xi}{g_{11}} \\ \psi &= \frac{1}{2} \left( \frac{(g_{22})_\eta}{g_{22}} - \frac{(g_{11})_\eta}{g_{11}} \right)\end{aligned}\tag{4.22}$$

$$\begin{aligned}\phi &= \frac{1}{2} \left( \frac{(g_{11})_\xi}{g_{11}} - \frac{(g_{22})_\xi}{g_{22}} \right) \\ \psi &= \frac{1}{2} \left( \frac{(g_{22})_\eta}{g_{22}} - \frac{(g_{11})_\eta}{g_{11}} \right) \\ \theta &= \frac{1}{2} \left( \frac{(g_{33})_\zeta}{g_{33}} - \frac{(g_{11})_\zeta}{g_{11}} - \frac{(g_{22})_\zeta}{g_{22}} \right)\end{aligned}\tag{4.23}$$

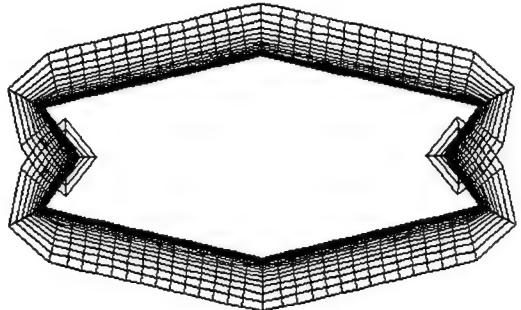
All control functions are computed from the original grid which may have grid line crossing somewhere in the grid. To ensure obtaining good forcing control functions,  $\phi$  for two dimensional grid, and  $\phi$  and  $\psi$  for three dimensional grid are computed at every level, but  $\psi$  for two dimensional grid, and  $\theta$  for three dimensional grid are only computed at the first level.

Smoothing the grids by solving the Poisson equations preserves orthogonality for grid lines inside the grids, and precise grid line concentration, which resolves the problem that concentration of the grid lines would be lost while smoothing the grids by solving the Laplacian equations.

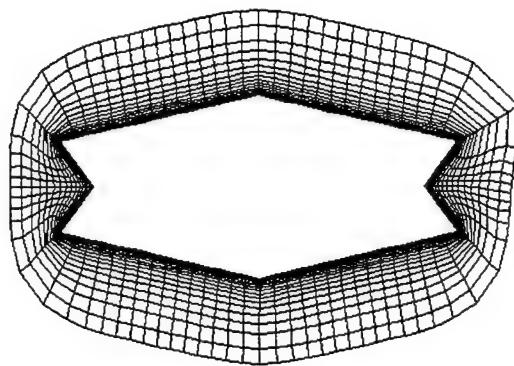
Figure 4.3, Figure 4.4, and Figure 4.5 illustrate the results of front advancing grid generation with elliptic smoothing and without elliptic smoothing. Also, phenomena of elliptic smoothing with forcing control functions and without forcing control functions are shown from Figure 4.6 to Figure 4.13.



**Figure 4.3 Effect of Local Elliptic Solver on Front Advancing Grid Generation.**

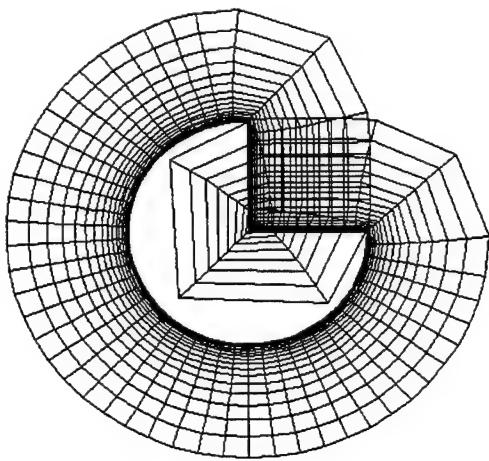


(a) Without Elliptic Smoothing.

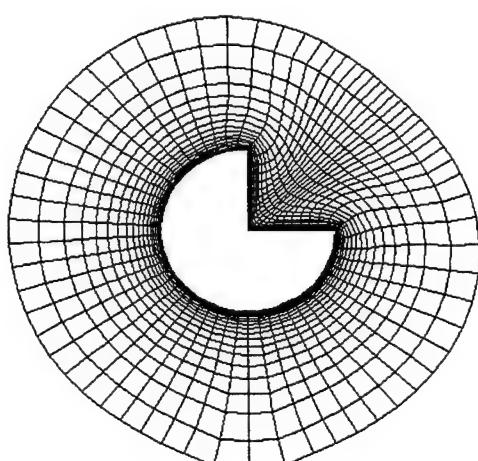


(b) With Elliptic Smoothing.

**Figure 4.4 Effect of Local Elliptic Solver on Front Advancing Grid Generation.**

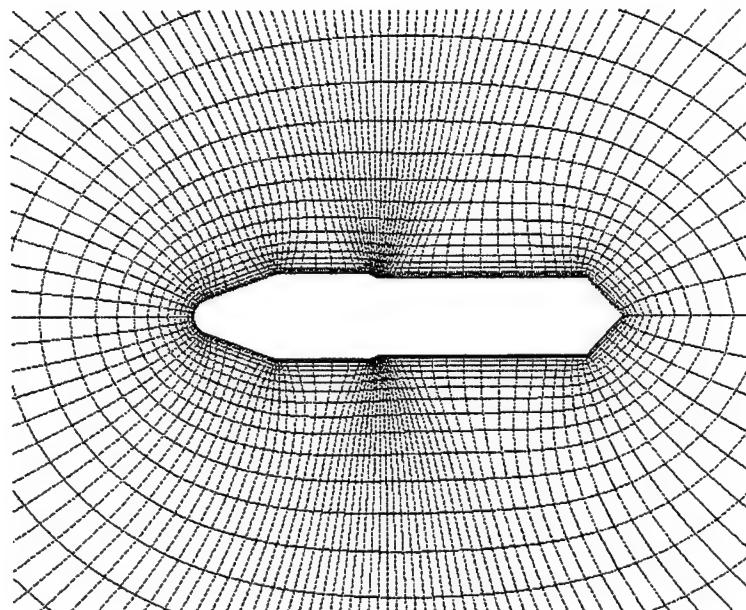


(a) Without Elliptic Smoothing.

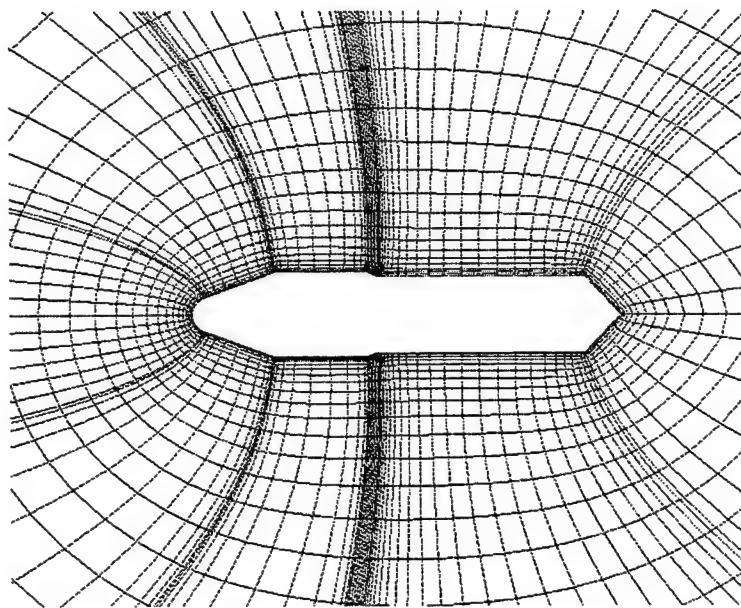


(b) With Elliptic Smoothing.

**Figure 4.5 Effect of Local Elliptic Solver on Front Advancing Grid Generation.**



**Figure 4.6 Rocket Grid With Zero Forcing Functions.**



**Figure 4.7 Rocket Grid With Non-Zero Forcing Functions.**

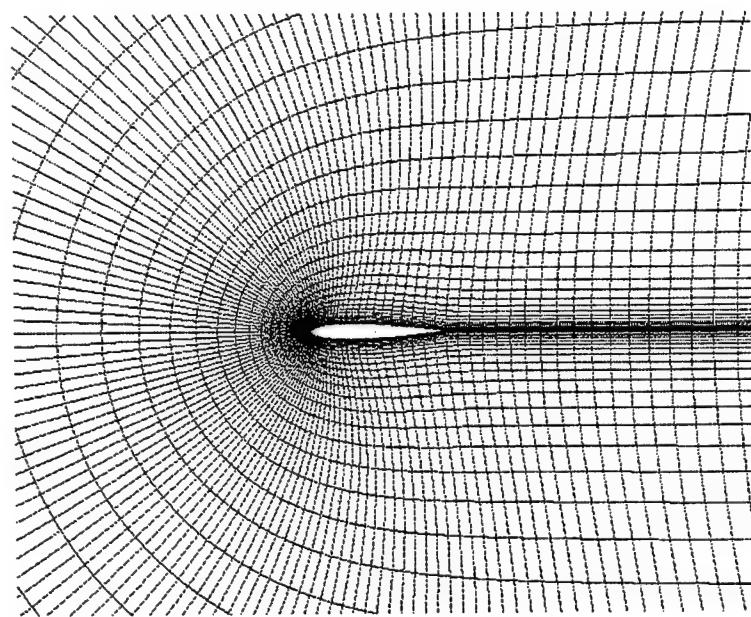


Figure 4.8 C-Type Airfoil Grid With Zero Forcing Functions.

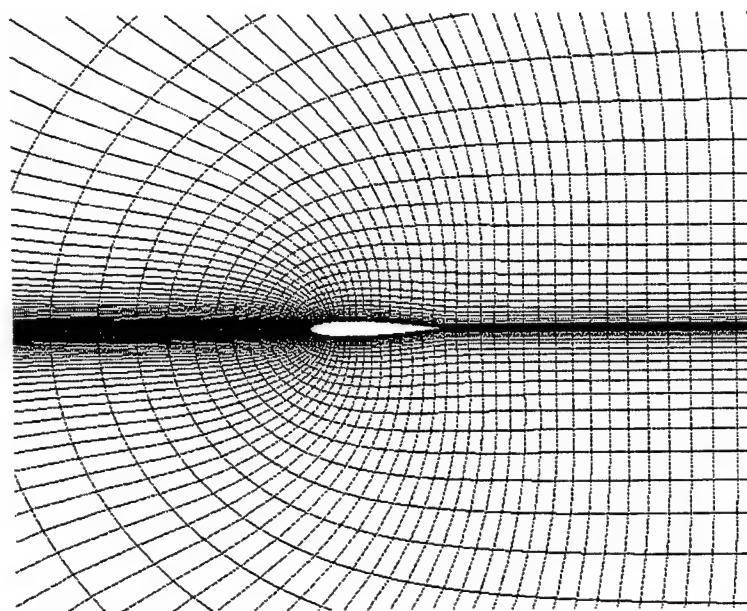


Figure 4.9 C-Type Airfoil Grid With Non-Zero Forcing Functions.

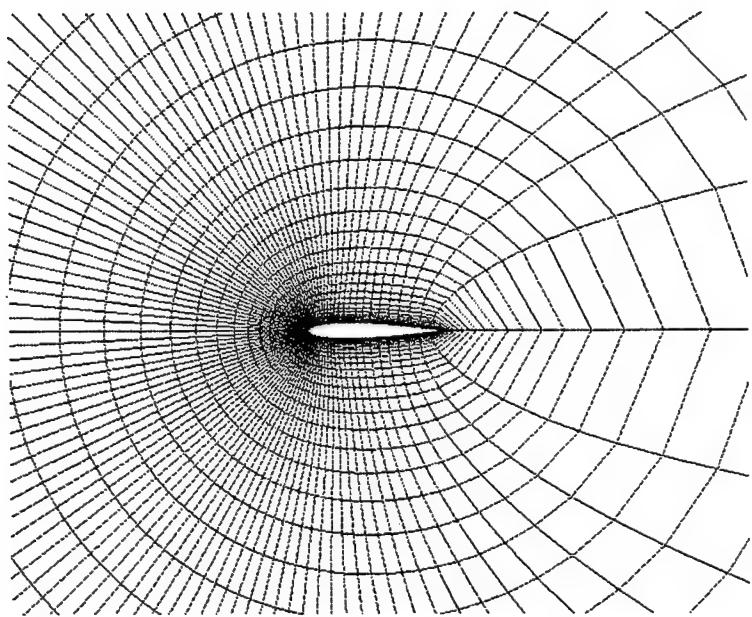


Figure 4.10 O–Type Airfoil Grid With Zero Forcing Functions.

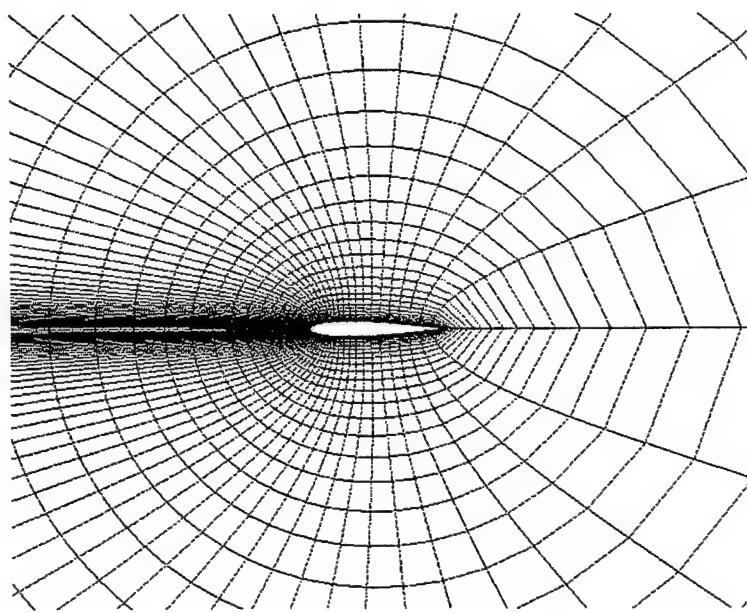
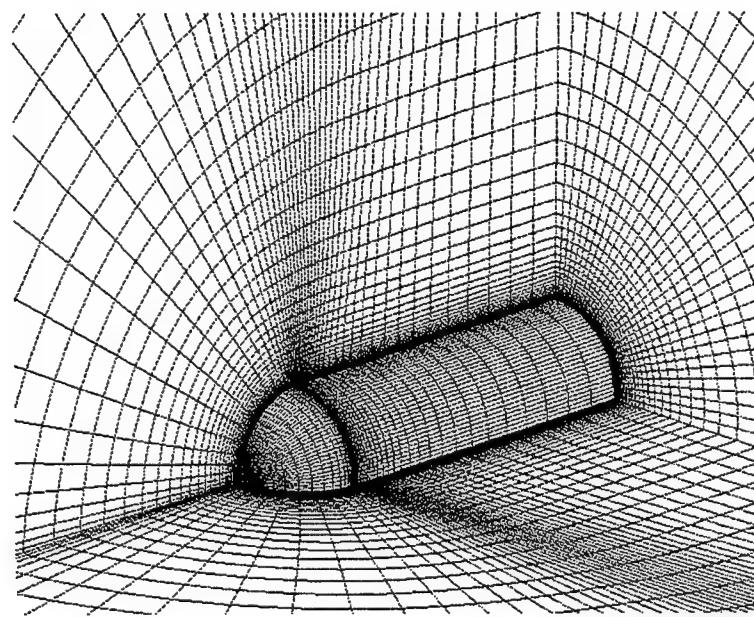
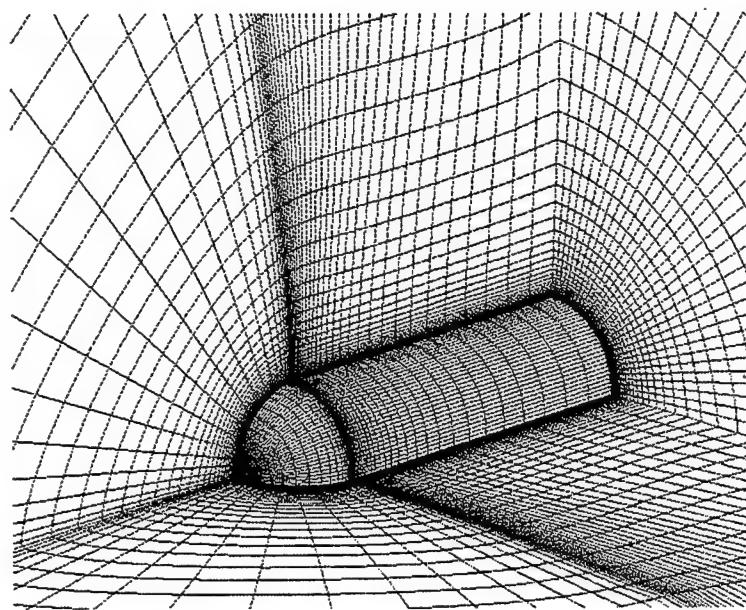


Figure 4.11 O–Type Airfoil Grid With Non–Zero Forcing Functions.



**Figure 4.12** Missile Grid With Zero Forcing Functions.



**Figure 4.13** Missile Grid With Non-Zero Forcing Functions.

#### 4.1.2 Boundary Conditions

Cut line boundary points are smoothed by two different approaches which are based on Bezier curve and liner extrapolation. Bezier curve is used for the boundary where the first point and the last point of the same grid line overlap together. Linear extrapolation is used to extrapolate the points from inside the grid to the boundary. Without specifying the boundary conditions, the concave boundary of the grid could cause the grid line crossing and ruin the whole volume grid.

##### 4.1.2.1 Bezier curve

A cubic polynomial is written as parametric form in equation (4.24). If we have the boundary conditions of a curve as in equation (4.25), we can rewrite the equation (4.24) to equation (4.26) which is Bezier curve of 3rd degree. By controlling the boundary conditions of the curve, we can easily smooth the points inside the curve for any given location controlled by parameter  $t$ .

$$r(t) = a + b t + c t^2 + d t^3 \quad (4.24)$$

$$\begin{aligned} r(0) &= C_1 \\ r(1) &= C_4 \\ r'(0) &= 3(C_2 - C_1) \\ r'(1) &= 3(C_4 - C_3) \end{aligned} \quad (4.25)$$

$$r(t) = \phi_1 C_1 + \phi_2 C_2 + \phi_3 C_3 + \phi_4 C_4 \quad (4.26)$$

where

$$\phi_1 = (1 - t)^3$$

$$\phi_2 = 3t(1 - t)^2$$

$$\phi_3 = 3t^2(1 - t)$$

$$\phi_4 = t^3$$

Figure 4.14, and Figure 4.15 show the Bezier curve smoothing effect.

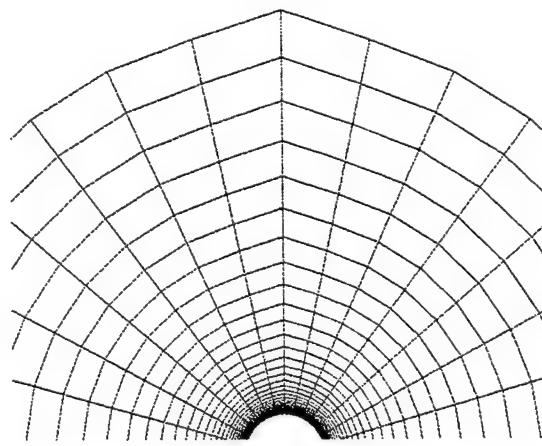


Figure 4.14 Boundary Conditions Without Smoothing.

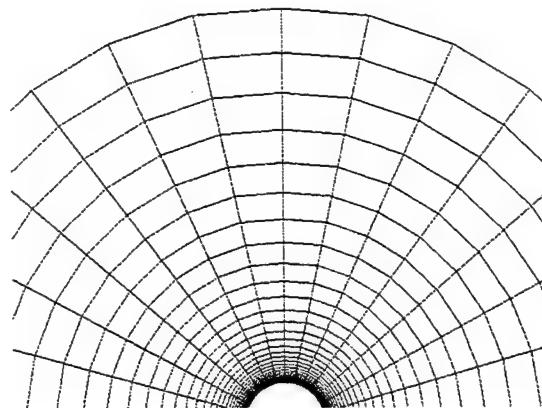


Figure 4.15 Boundary Conditions With Bezier Curve Smoothing.

#### 4.1.2.2 Linear Extrapolation

Linear extrapolation is performed by solving the intersection of a line from inside the grid and a boundary surface. The line and surface can be written as equations (4.27) and (4.28) respectively. By solving these equations, we can get the point,  $r$ , on the boundary surface.

$$r - r_1 = m \vec{V} \quad (4.27)$$

$$(r - r_0) \cdot \vec{e} = 0 \quad (4.28)$$

where

$r_0$  : known point on the boundary surface

$r_1$  : known point on the grid line

$\vec{e}$  : normal vector on the surface

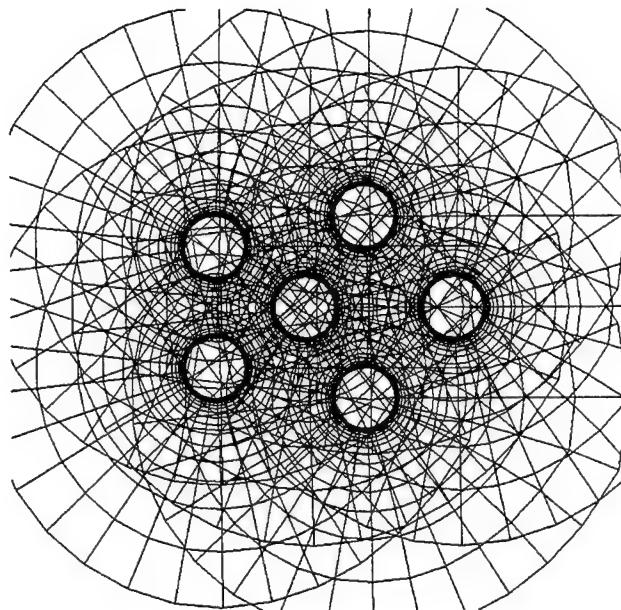
$\vec{V}$  : vector of the grid line

$m$  : unknown parameter of line equation

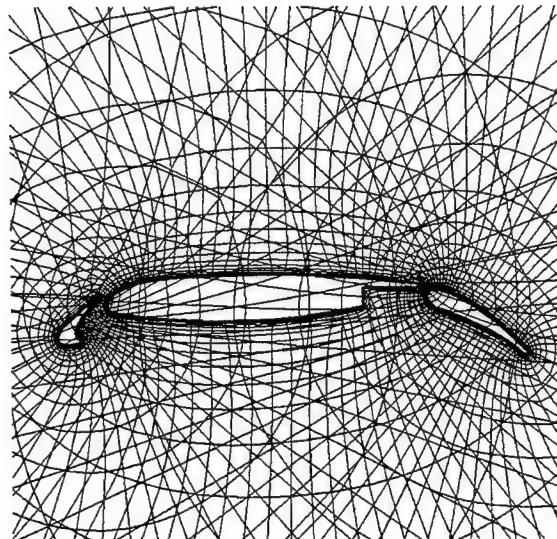
#### 4.2 Grid Trimming

Structured grids around the bodies are generated independently. All individual structured grids are then supplied to the hybrid grid generation system for trimming the overlapped structured grid. Some overlapped structured grids are shown in Figure 4.16 through Figure 4.18. Figure 4.16 shows that six cylinder grids are generated and overlap with each others. Figure 4.17 and Figure 4.18 show the three element airfoil structured grids and two element airfoil structured grids respectively.

Grids might overlap with others without the control of marching distances of each individual process. So overlapping cells removal is the most important concept of this grid trimming work. For an optimal interface between quadrilateral cells and triangular cells, aspect ra-

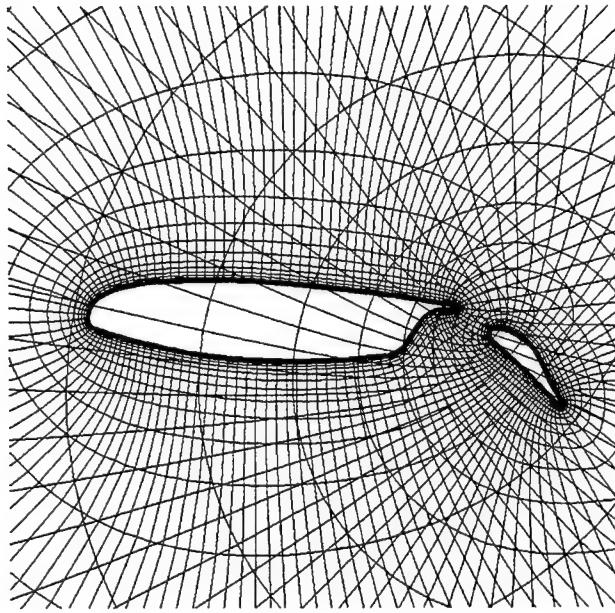


**Figure 4.16 Six Cylinder Grids Overlap With Each Other.**



**Figure 4.17 Three Element Airfoil Grids Overlap With Each Other.**

tios of quadrilateral cells are also checked. The three different steps in trimming the overlapped structured grids are the following. First, overlapping cells from different sets of structured grids are removed. Second, aspect ratios of remaining cells are checked. Third, boundary



**Figure 4.18 Two Element Airfoil Grids Overlap With Each Other.**

points are enriched if the distances between two boundaries are not enough for optimal aspect ratio cells wrapping around the grids.

Overlapping cells removal process is based on the comparison of distances between nodes of cells from different set of structured grids and solid boundaries. This process is the initial step of grid trimming work, which only ensures that all overlapping cells are removed and all individual grids are separated from others by a clear area, or a void area. An aspect ratio checking process then performs to remove all cells which does have an aspect ratio less than unity. High aspect ratio cells are kept around individual solid boundaries for efficient viscous and turbulent flow simulations (Marcum [39]). Low aspect ratio cells far away from the solid boundaries are removed. Because low aspect ratio cells cause bad grid interface between quadrilateral and triangular cells, which affect the smooth area transition of the cell in structured to unstructured region.

Boundary points are enriched locally where the distance between two solid boundaries is not far enough to have perfect aspect ratio cells wrapping around the boundaries (Marcum

[39]). If a region between two boundaries is very narrow, the local flow solution is very critical to the global one. Figure 4.19 to Figure 4.22 show the comparison of the grid trimming with and without aspect ratio checking. Figure 4.23 and Figure 4.24 illustrate the trimmed grids of two-element and four-element airfoils. Boundary point enrichment effect are shown in Figure 4.25 to Figure 4.32.

The void between the trimmed structured grids are filled with triangles. Delaunay triangulation is used to generate triangular cells.

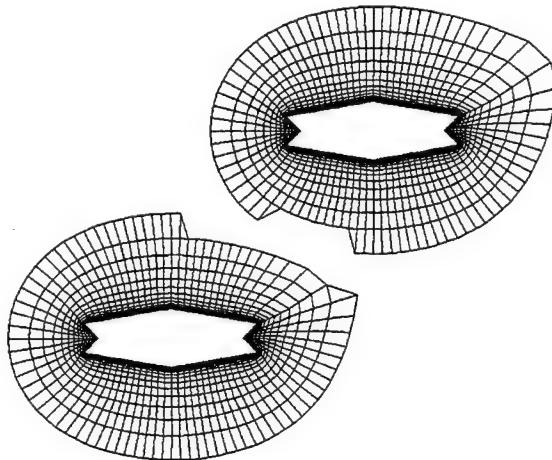


Figure 4.19 Grid Trimming Without Aspect Ratio Checking.

Results of two dimensional and three dimensional grids utilizing front advancing and elliptic PDE smoothing grid generator are shown below from Figure 4.33 to Figure 4.40.

Hybrid grid around multiple cylinders and around two arbitrary geometries hybrid grids are shown in Figure 4.41 and Figure 4.42 respectively. Hybrid grids around multi-element airfoils of practical importance to aerospace engineering applications are shown in Figure 4.43 to Figure 4.45.

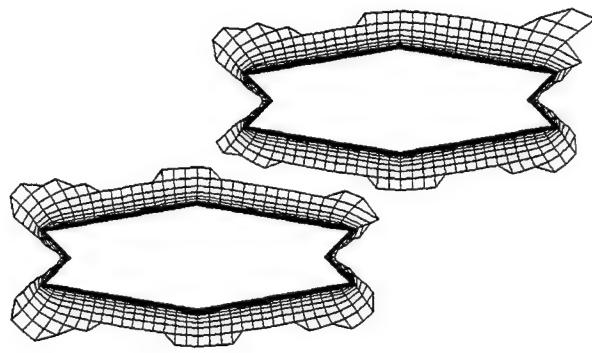


Figure 4.20 Grid Trimming With Aspect Ratio Checking.

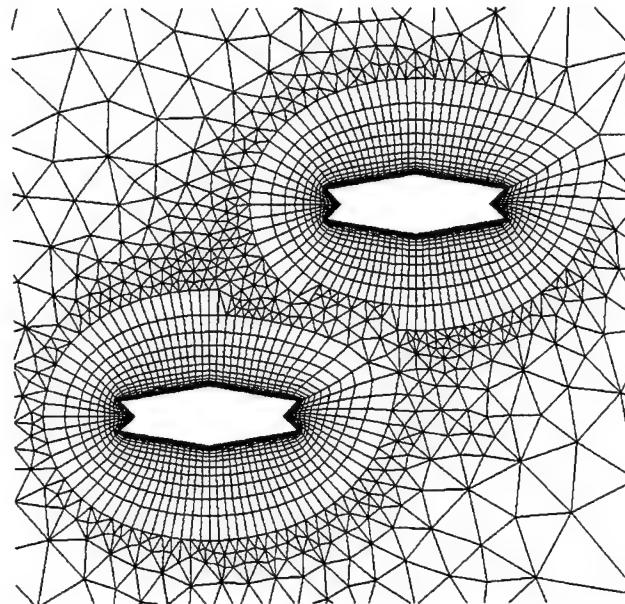
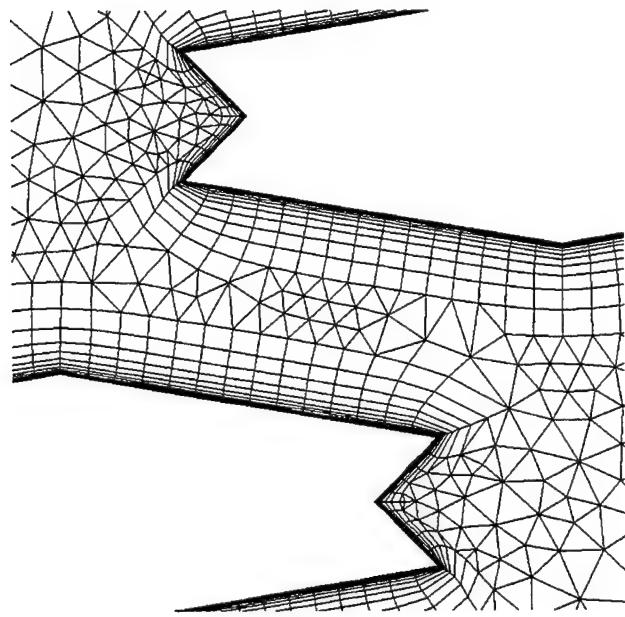
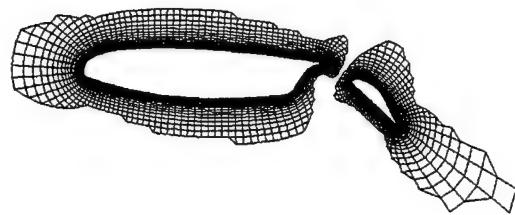


Figure 4.21 Grid Quality Is Not Good Without Aspect Ratio Checking.



**Figure 4.22 Grid Quality Is Improved With Aspect Ratio Checking.**



**Figure 4.23 Trimmed Grids For Two-Element Airfoil.**

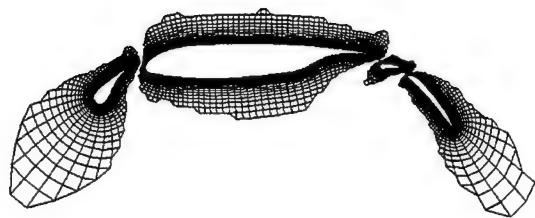


Figure 4.24 Trimmed Grids For Four-Element Airfoil.

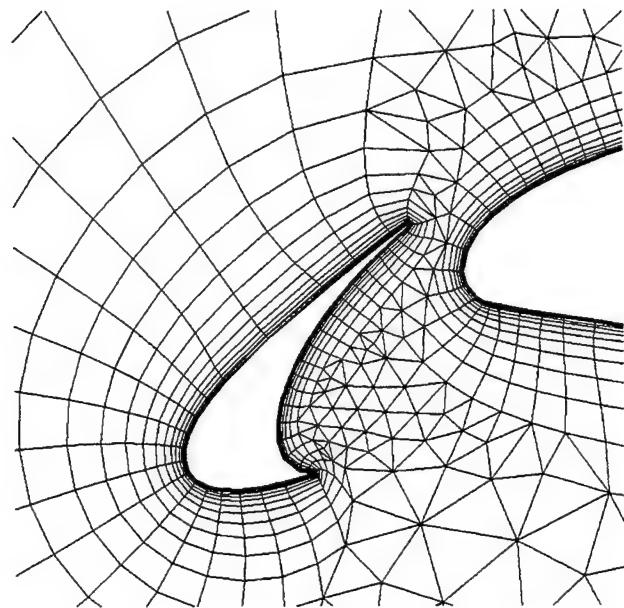


Figure 4.25 Grid Trimming Without Boundary Point Enrichment.

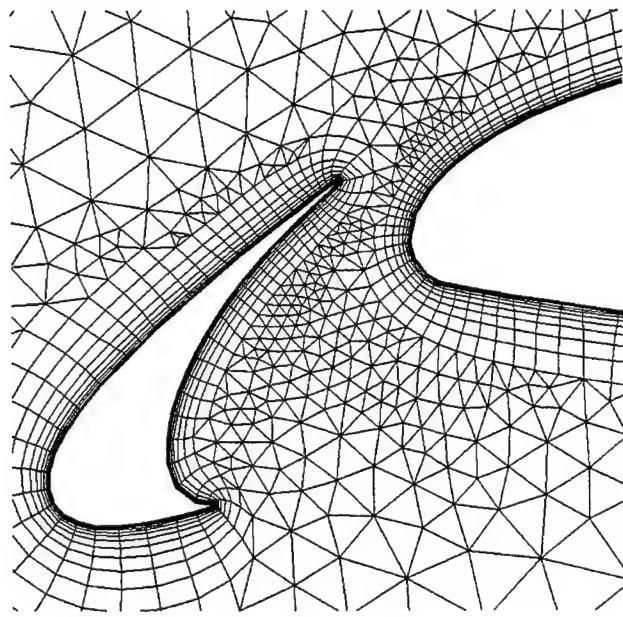


Figure 4.26 Grid Trimming With Boundary Point Enrichment.

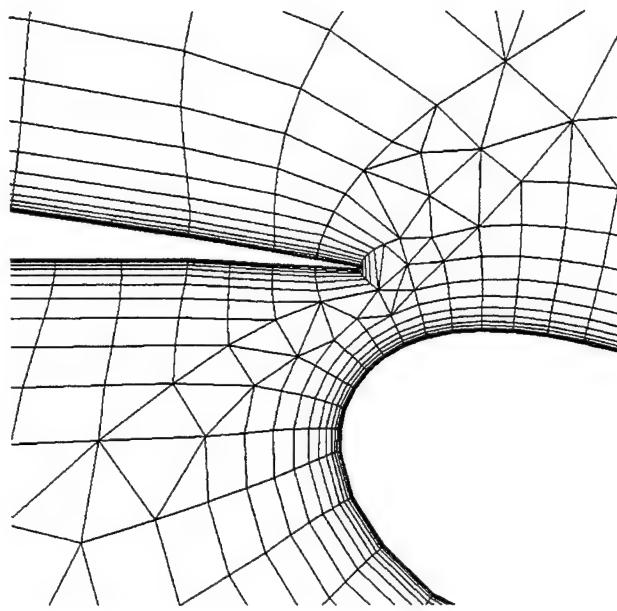


Figure 4.27 Grid Trimming Without Boundary Point Enrichment.

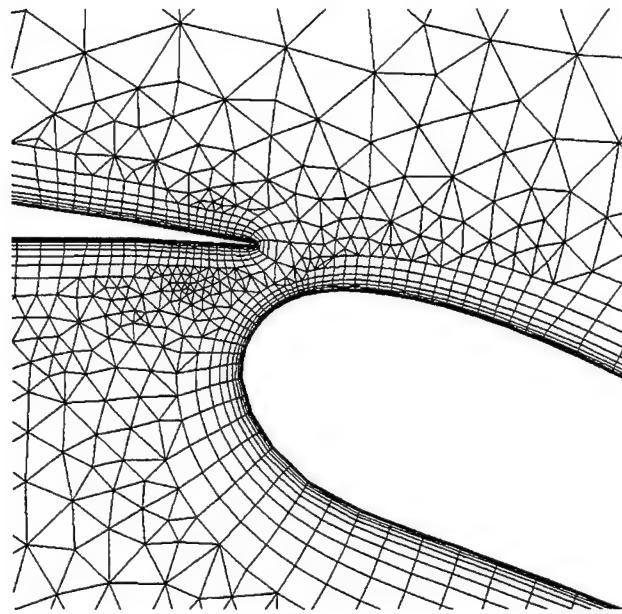


Figure 4.28 Grid Trimming With Boundary Point Enrichment.

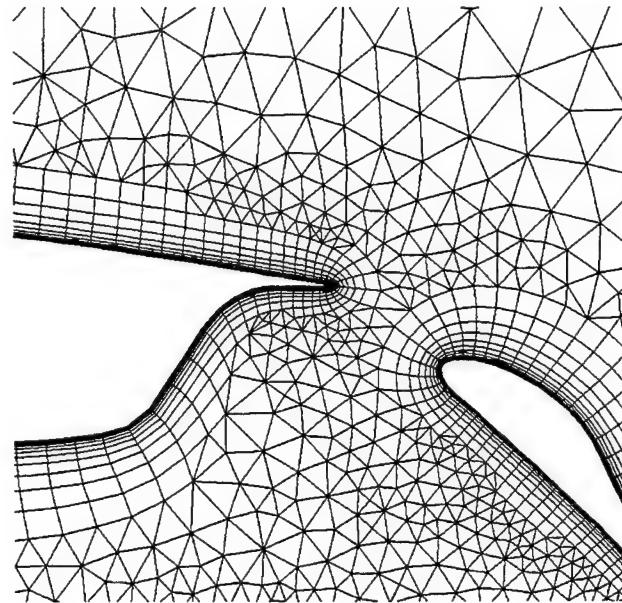


Figure 4.29 Grid Trimming Without Boundary Point Enrichment.

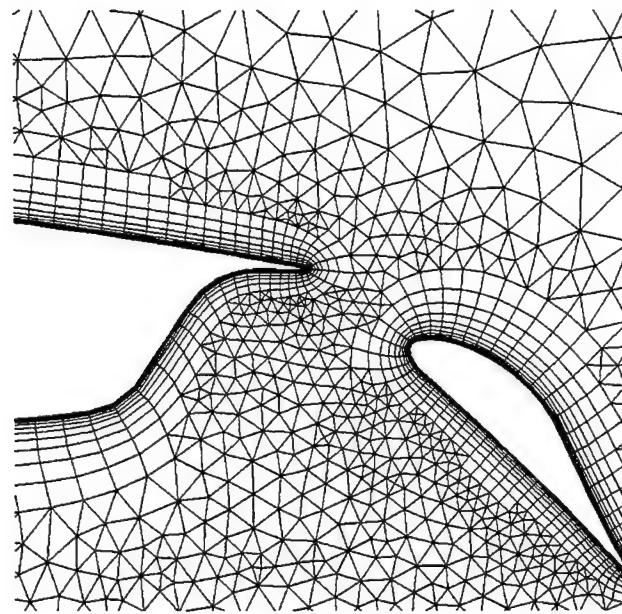


Figure 4.30 Grid Trimming With Boundary Point Enrichment.

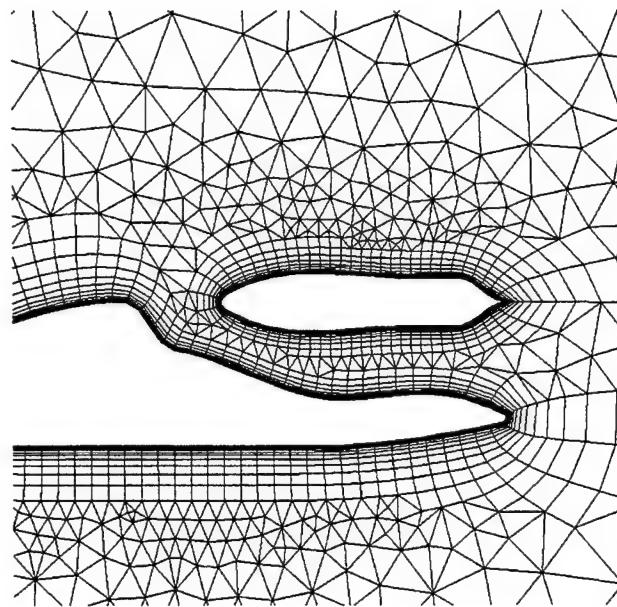


Figure 4.31 Grid Trimming Without Boundary Point Enrichment.

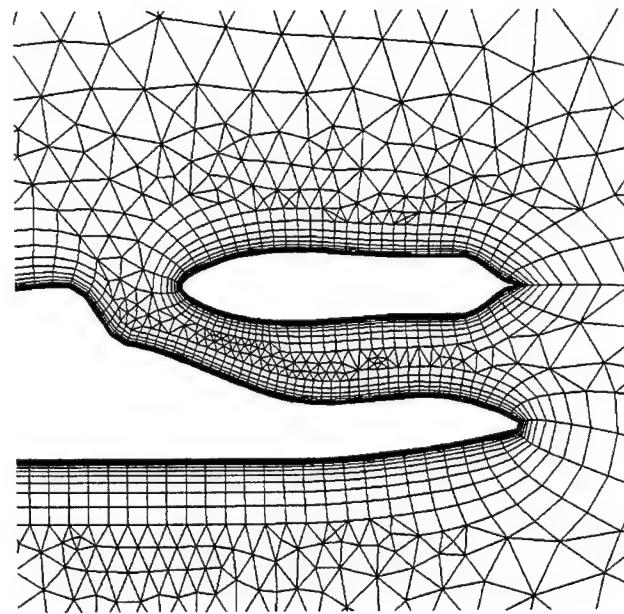


Figure 4.32 Grid Trimming With Boundary Point Enrichment.

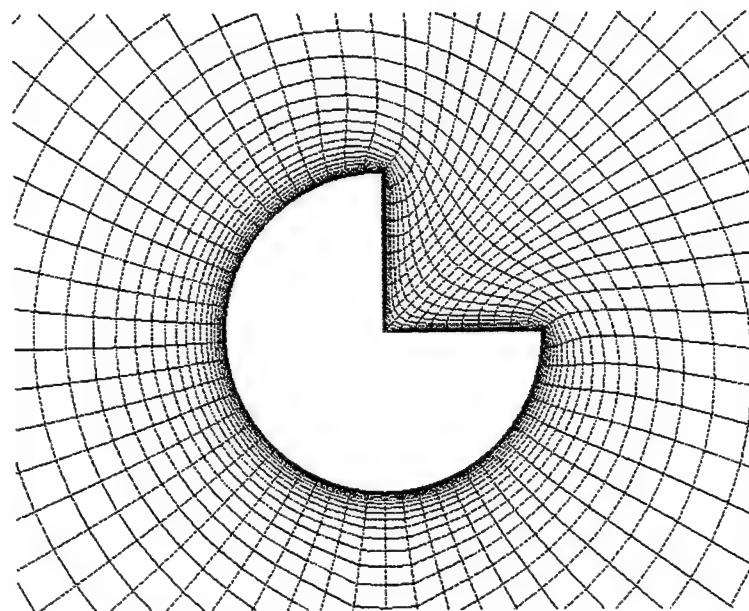


Figure 4.33 3/4 Circle Grid With Forcing Functions.

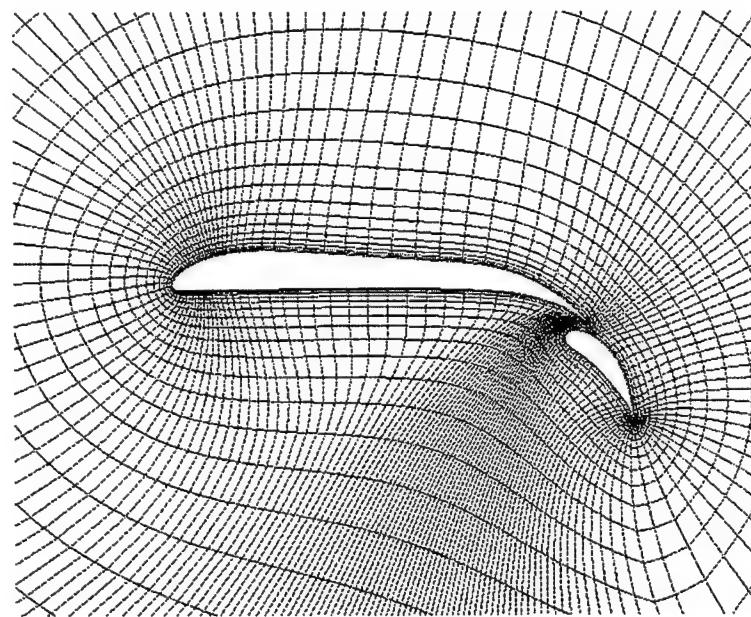


Figure 4.34 O–Type Two–Element Airfoil Grid.

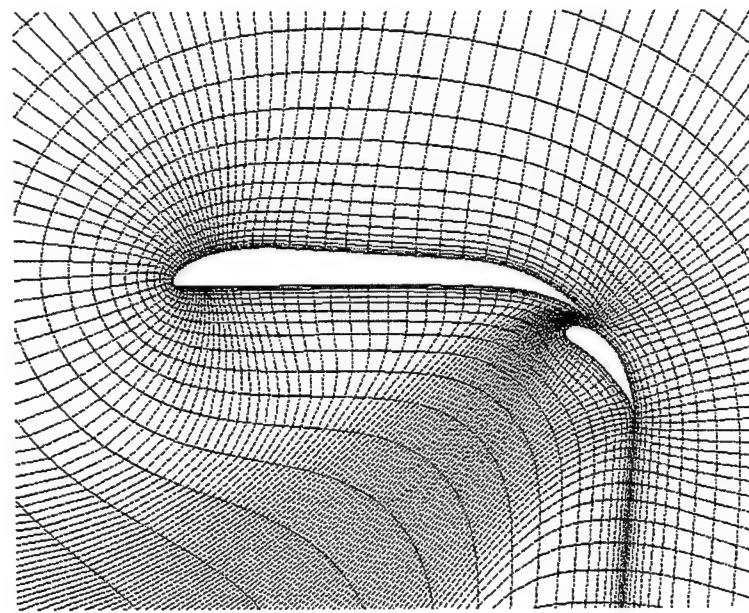


Figure 4.35 C–Type Two–Element Airfoil Grid.

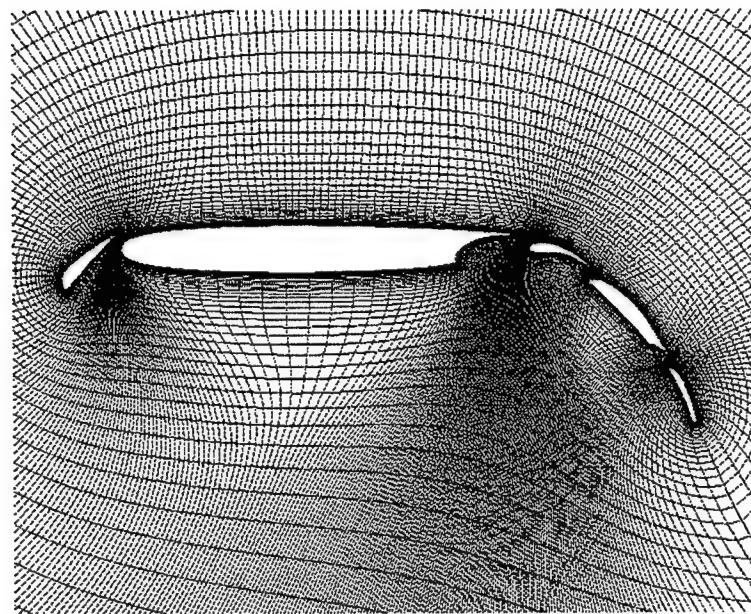


Figure 4.36 Five-Element Airfoil Grid.

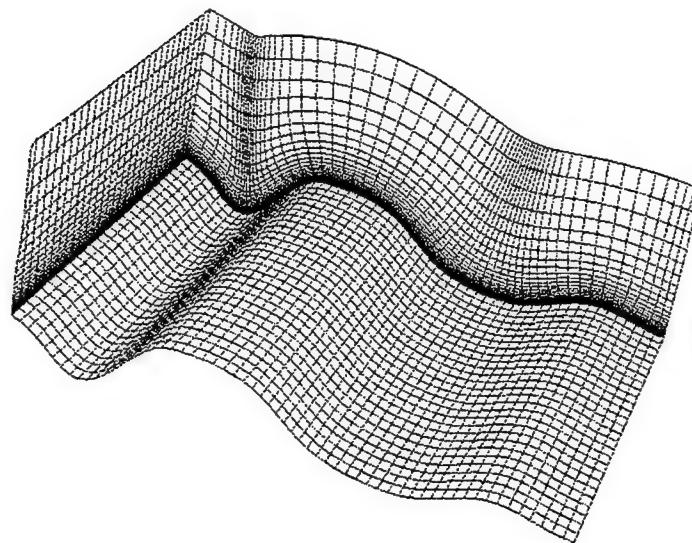


Figure 4.37 Three Dimensional Surface Grid.

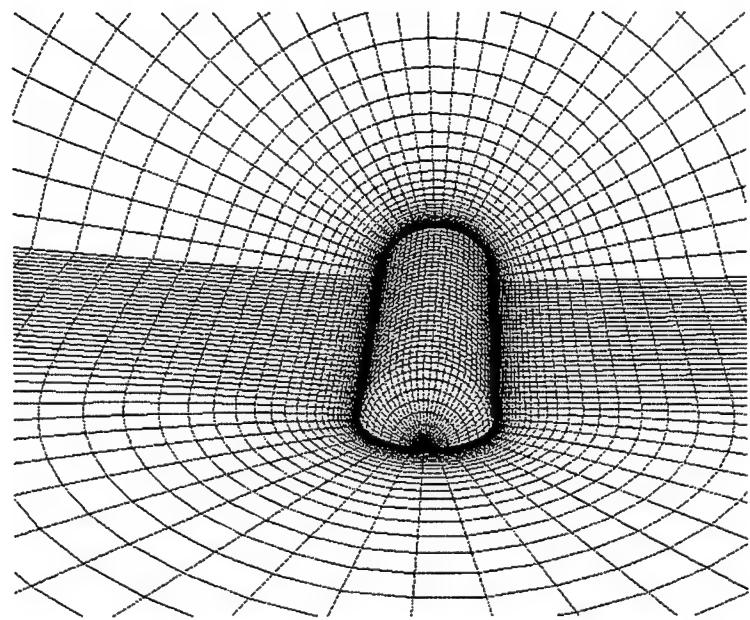


Figure 4.38 Three Dimensional Missile Grid.

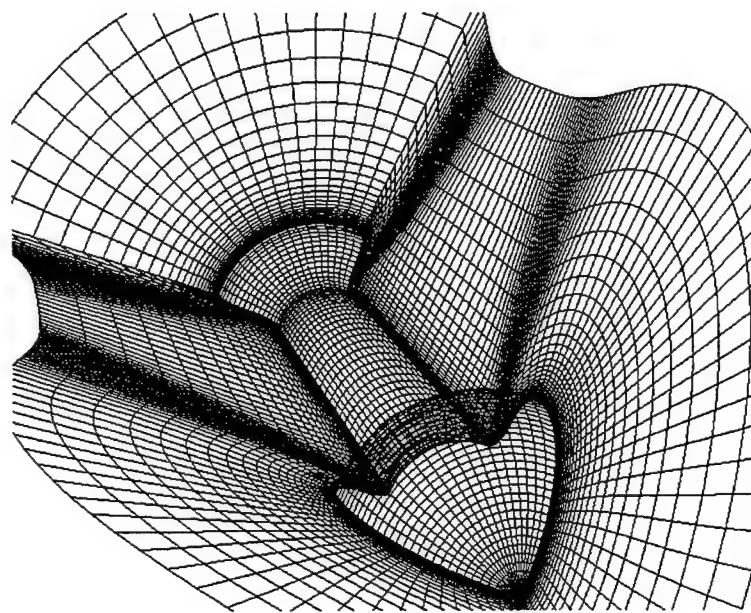


Figure 4.39 Three Dimensional Arbitrary Geometry Grid.

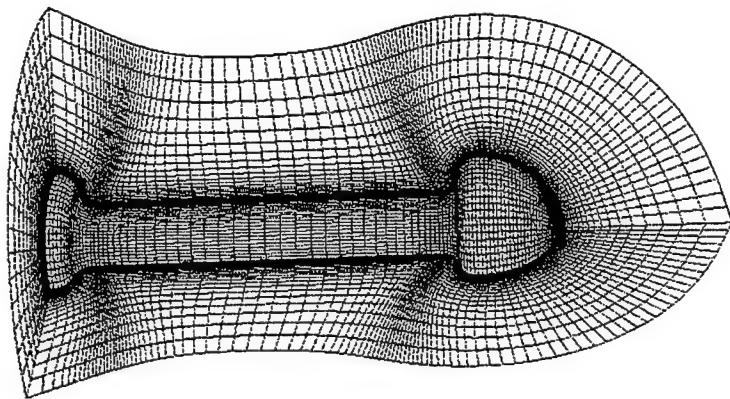


Figure 4.40 Three Dimensional Arbitrary Geometry Grid.

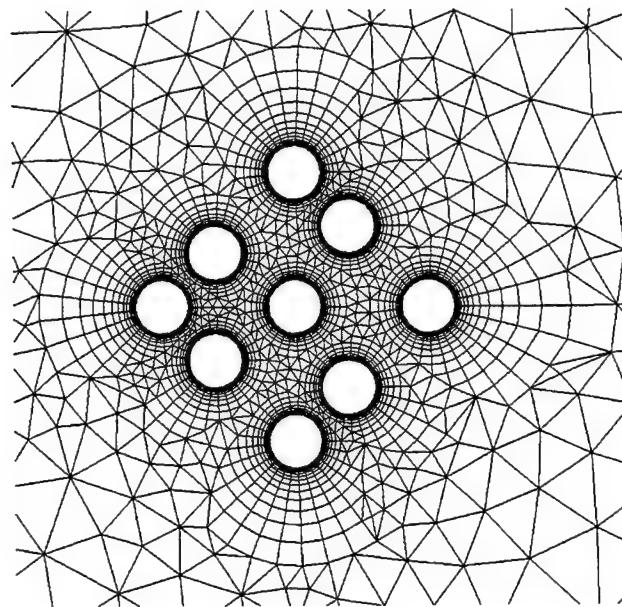


Figure 4.41 Nine Cylinder Hybrid Grid.

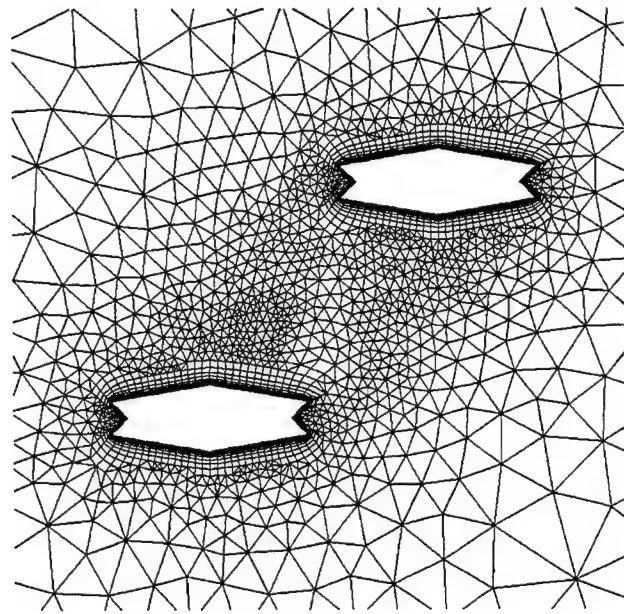


Figure 4.42 Two Arbitrary Geometries Hybrid Grid.

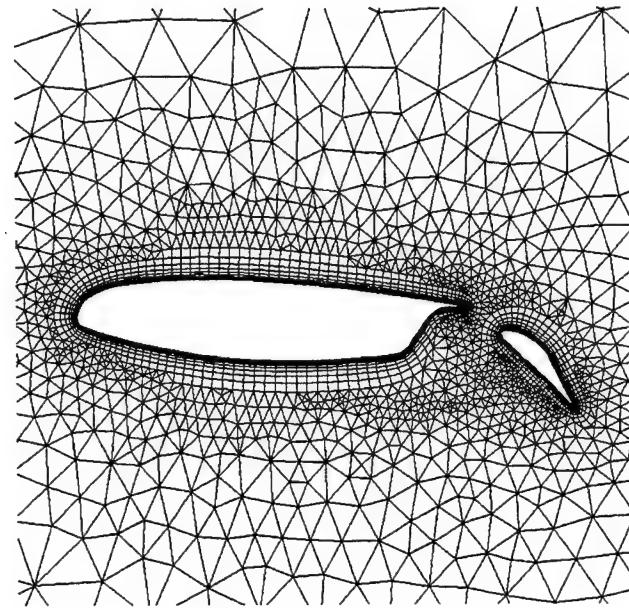


Figure 4.43 Two Element Airfoil Hybrid Grid.

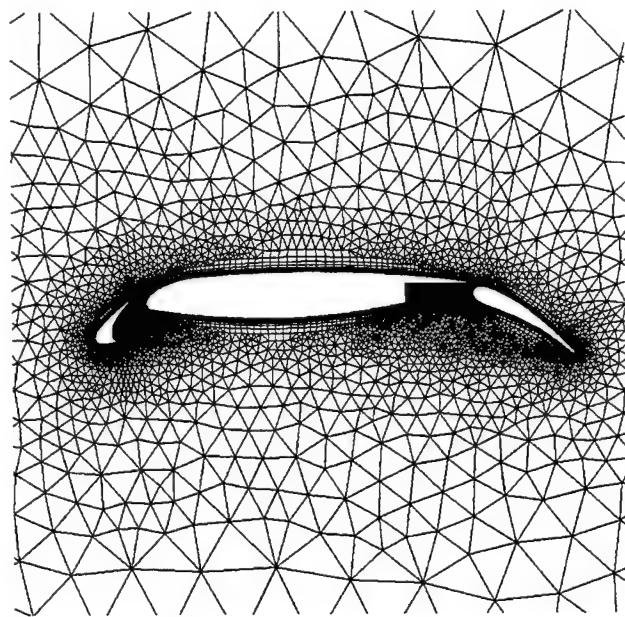


Figure 4.44 Three Element Airfoil Hybrid Grid.

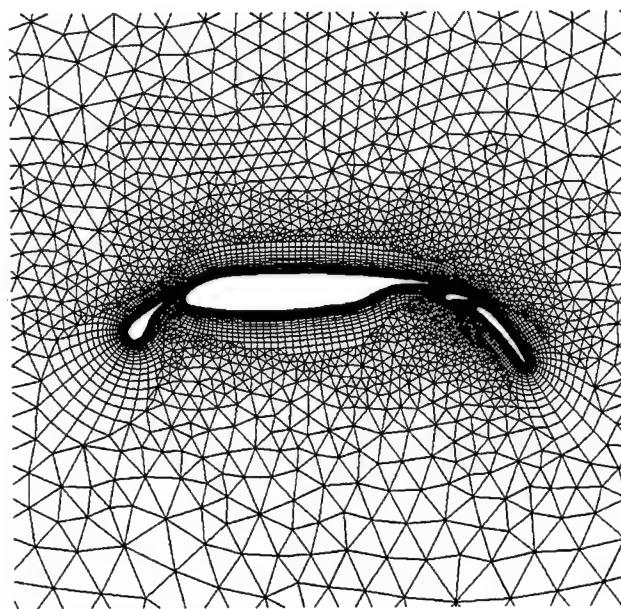


Figure 4.45 Four Element Airfoil Hybrid Grid.

## 5. Hybrid Flow Solver

A hybrid grid is a combination of structured and unstructured grids. The number of edges (2D) or faces (3D) that form a cell can vary from one cell to another. The arbitrariness in the number of faces for a given cell makes the finite volume method the natural choice for solving the governing equations. The finite volume schemes are based on a discretization of the integral form of the equations. The governing equations used in the present study are the Navier–Stokes equations. The following section presents the Navier–Stokes equations in the integral form.

### 5.1 Governing Equations

The non-dimensionalizations are done based on freestream conditions. The characteristic velocity is taken as the freestream velocity. Neglecting the body forces, the non-dimensional form of the Navier–Stokes equations can be written in vector notation as,

$$\frac{\partial}{\partial t} \int_{\Omega} Q d\Omega + \oint_{\partial\Omega} \underline{F}(Q) \cdot \underline{n} ds = \oint_{\partial\Omega} \underline{F}^v(Q) \cdot \underline{n} ds \quad (5.1)$$

$$\text{where } \underline{F}(Q) \cdot \underline{n} ds = \mathcal{F} = (f(Q)\underline{i} + g(Q)\underline{j} + h(Q)\underline{k}) \cdot \underline{n} ds \quad (5.2)$$

$$\underline{F}^v(Q) \cdot \underline{n} ds = \mathcal{F}^v = (f^v(Q)\underline{i} + g^v(Q)\underline{j} + h^v(Q)\underline{k}) \cdot \underline{n} ds \quad (5.3)$$

$$\underline{n} = n_x \underline{i} + n_y \underline{j} + n_z \underline{k} \quad (5.4)$$

The vector  $\underline{n}$  is the outward unit normal to the control surface with  $n_x$ ,  $n_y$ , and  $n_z$  as the components in  $x$ ,  $y$ , and  $z$  directions and  $ds$  is the cell face area for 3-dimensions and the edge length in 2-dimensions. The conserved variables  $Q$ , the convective flux vectors  $f$ ,  $g$ , and  $h$  and the viscous flux vectors  $f^v$ ,  $g^v$ , and  $h^v$  are defined as,

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix} \quad f = \begin{bmatrix} \rho U \\ \rho U u + p \\ \rho U v \\ \rho U w \\ U(E + p) - x_t p \end{bmatrix}$$

$$g = \begin{bmatrix} \rho V \\ \rho u V \\ \rho V v + p \\ \rho V w \\ V(E + p) - y_t p \end{bmatrix} \quad h = \begin{bmatrix} \rho W \\ \rho u W \\ \rho v W \\ \rho W w + p \\ W(E + p) - z_t p \end{bmatrix}$$

$$f^v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix} \quad g^v = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{xy} + v\tau_{yy} + w\tau_{yz} - q_y \end{bmatrix}$$

$$h^v = \begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - q_z \end{bmatrix}$$

where  $E$  is the total energy per unit volume and is defined as,

$$E = \rho e_t = \frac{p}{\gamma - 1} + \rho \left( \frac{u^2 + v^2 + w^2}{2} \right) \quad (5.5)$$

The grid speeds in the  $x$ ,  $y$ , and  $z$  co-ordinate directions are  $x_t$ ,  $y_t$ , and  $z_t$  respectively and the variables  $U$ ,  $V$  and  $W$  are defined as,  $u = x_t$ ,  $v = y_t$ , and  $w = z_t$  respectively. The ratio of specific heats  $\gamma$  for standard air is taken as 1.4.

The components of viscous shear stress tensor  $\tau_{ij}$  represents the shear stress acting in a plane of constant  $i$  in the direction of  $j$ , with  $i$  and  $j$  as either  $x$ ,  $y$ , or  $z$ . The heat transfer by conduction in the  $i$  direction is represented as  $q_i$ . The components of viscous shear stress tensor  $\tau$ , and the heat transfer  $q$  are defined as,

$$\begin{aligned}
\tau_{xx} &= \frac{2(\mu + \mu_t)}{3 \text{Re}_L} \left( 2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right) \\
\tau_{yy} &= \frac{2(\mu + \mu_t)}{3 \text{Re}_L} \left( 2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right) \\
\tau_{zz} &= \frac{2(\mu + \mu_t)}{3 \text{Re}_L} \left( 2 \frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) \\
\tau_{xy} &= \frac{(\mu + \mu_t)}{\text{Re}_L} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\
\tau_{xz} &= \frac{(\mu + \mu_t)}{\text{Re}_L} \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\
\tau_{yz} &= \frac{(\mu + \mu_t)}{\text{Re}_L} \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \\
q_x &= - \frac{1}{(\gamma - 1) M_\infty^2 \text{Re}_L} \left( \frac{\mu}{\text{Pr}} + \frac{\mu_t}{\text{Pr}_t} \right) \frac{\partial T}{\partial x} \\
q_y &= - \frac{1}{(\gamma - 1) M_\infty^2 \text{Re}_L} \left( \frac{\mu}{\text{Pr}} + \frac{\mu_t}{\text{Pr}_t} \right) \frac{\partial T}{\partial y} \\
q_z &= - \frac{1}{(\gamma - 1) M_\infty^2 \text{Re}_L} \left( \frac{\mu}{\text{Pr}} + \frac{\mu_t}{\text{Pr}_t} \right) \frac{\partial T}{\partial z}
\end{aligned}$$

where  $M_\infty$  is the freestream Mach number. The turbulence viscosity  $\mu_t$  is calculated using the Spalart–Allmaras one equation turbulence model ( Spalart and Allmaras [35] ). The turbulent Prandtl number  $\text{Pr}_t$  is taken as 0.92.

The nondimensional form of the equation of state for an ideal gas is written as,

$$p = \frac{\rho T}{\gamma M_\infty^2} \quad (5.6)$$

The nondimensional laminar viscosity coefficient is given by the relation

$$\mu = \frac{(1 + c_3)T^{3/2}}{T + c_3} \quad (5.7)$$

where  $c_3 = \frac{c_2}{\hat{T}_\infty}$  and the reference temperature  $\hat{T}_\infty$  is taken as 300 K.

## 5.2 Cell Face-Based Data Structure

Hybrid grids consist of polygons with an arbitrary number of sides. This makes it necessary to use a generalized data structure to store the grid information. The grid information can be stored in two different ways: (i) Cell-based data structure (ii) Cell face-based data structure. These two approaches are best explained by an example.

Consider a two-dimensional grid that consists of two cells, one with five sides and the other with three sides (Figure 5.1). The nodes are denoted by  $N_1, N_2, \dots, N_6$ , the cells by  $C_1, C_2$  and the edges by  $e_1, e_2, \dots, e_7$ . In the cell-based data structure the grid is represented as shown in Table 5.1. The nodes that form a cell are usually ordered in either a clockwise or a counter-clockwise direction. In this case, the counter-clockwise ordering of nodes was chosen.

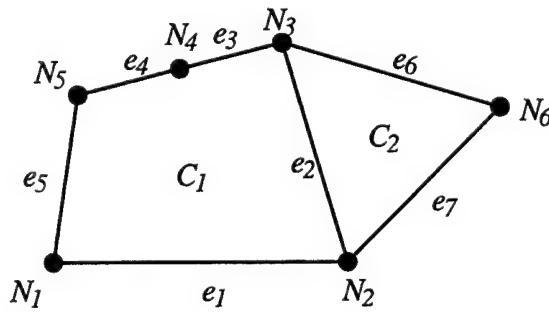


Figure 5.1 Example of Hybrid Grid

Table 5.1 Cell Based Data Structure

| Cell No. | No. of Nodes | Node 1 | Node 2 | Node 3 | Node 4 | Node 5 |
|----------|--------------|--------|--------|--------|--------|--------|
| $C_1$    | 5            | $N_1$  | $N_2$  | $N_3$  | $N_4$  | $N_5$  |
| $C_2$    | 3            | $N_2$  | $N_6$  | $N_3$  |        |        |

In 2-dimensional cases the cell face-based data structure will change to an edge-based one. For the edge-based data structure, every edge has four pieces of information associated with it. They are the nodes that form the edge and the cell numbers on either side of the edge. For the boundary faces, the boundary condition information is stored in place of the cell number on the right side. The left and right sides of an edge (2D) or a face (3D) are defined in the

following manner. When one traverses an edge from the first node to the second node, the cell on the left hand side is denoted as the left cell and the cell on the other side as the right cell. In three dimensions, the nodes that form the cell faces are ordered in such a manner that the numbering is counter-clockwise with respect to one cell and clockwise with respect to the other. The cell with respect to which the node numbering is counter-clockwise is taken as the left cell. In the edge-based data structure, the grid given in Figure 5.1 is represented as shown in Table 5.2.

Table 5.2 Edge-Based Data Structure

| Edge Number | First Node | Second Node | Cell on left | Cell on right |
|-------------|------------|-------------|--------------|---------------|
| $e_1$       | $N_1$      | $N_2$       | $C_1$        | $bc$          |
| $e_2$       | $N_2$      | $N_3$       | $C_1$        | $C_2$         |
| $e_3$       | $N_3$      | $N_4$       | $C_1$        | $bc$          |
| $e_4$       | $N_4$      | $N_5$       | $C_1$        | $bc$          |
| $e_5$       | $N_5$      | $N_1$       | $C_1$        | $bc$          |
| $e_6$       | $N_6$      | $N_3$       | $C_2$        | $bc$          |
| $e_7$       | $N_2$      | $N_6$       | $C_2$        | $bc$          |

In Table 5.2,  $bc$  represents the index for boundary conditions.

The advantage of the edge based system is that it is independent of the number of edges/faces of the cells. Also, the integration over the control surfaces is made easier by employing this type of data structure. This can be better explained by a method for evaluating the area of the cells. The area of a cell can be estimated by the simple integral as,

$$Area = \oint_{\partial Q} x \, dy = \sum_{i=1}^k x_i \, dy_i \quad (5.8)$$

where  $k$  is the number of cell edges. Using the edge-based data structure, the above integral can be evaluated for the entire grid using the following procedure.

*Initialize all areas to zero*

*Loop over all edges*

*Calculate the average value of x coordinate,  $xe$*

*Calculate the change in y coordinate,  $dy$*

*Area of cell on left = Area of cell on left +  $xe * dy$*

*Area of cell on right = Area of cell on right -  $xe * dy$*

*end loop.*

The same principle can be applied for flux evaluation too. Once the the flux is evaluated for a face, appropriate contributions can be added to the cells on either side of the face.

### 5.3 Spatial Discretization

Finite volume ( Barth [42], Whitakar [43] ) and finite element based schemes ( Jameson et al. [44], Marcum and Weatherill [2], Morgan et al. [5] ) are the most common approaches used for solving the governing equations on unstructured grids. Finite volume schemes are best suited for hybrid grids because a typical hybrid grid is an agglomeration of polygons with different number of sides per polygon. There are basically two different approaches for storing the conserved variables, one is a node based approach while the other is a cell based approach. Figure 5.2 shows the areas (shaded region) that are used for estimating the averaged values of the conserved variables for the two different approaches. In the node based method the area considered for this averaging is the area around the nodes. This area is termed dual area since this is considered a dual to the area of the cell. For two dimensional cases this area is taken as the area enclosed by the polygon formed by connecting the cell centers of the neighboring cells to the midpoint of the corresponding edges. In the second approach, the values stored at the cell center are taken as the cell averaged values. The area for the averaging is taken as that of the cell itself.

In the present study, the cell centered approach is used for storing the conserved variables and for the subsequent solution procedure. The cell averaged flow variables at the cell center are calculated as an integral average, i.e.,

$$Q(C_i) = \frac{1}{V_i} \int_{\Omega} Q(x, y, z) d\Omega \quad (5.9)$$

where  $C_i$  is the cell number and  $V_i$  is the cell volume.

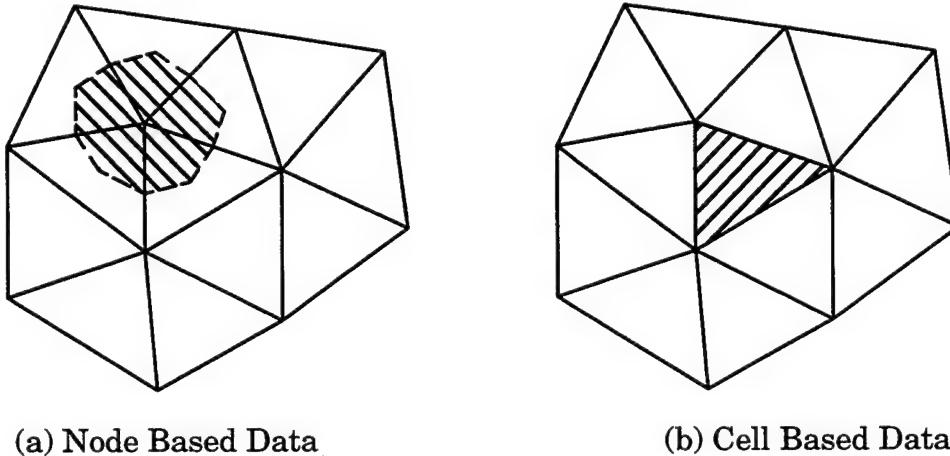


Figure 5.2 Areas Considered for Averaging the Conserved Variable in Node and Cell Based Schemes

### 5.3.1 Discretization of Convective Terms

The governing equations, equation (5.1), without the viscous terms are,

$$\frac{\partial}{\partial t} \int_{\Omega_i} Q_i d\Omega_i + \oint_{\partial\Omega} \underline{F}(Q) \cdot \underline{n} ds = 0 \quad (5.10)$$

Using the cell centered finite volume approach, a semi-discretized form of Equation (5.10) is

$$\frac{\partial}{\partial t} \int_{\Omega} Q d\Omega = - \sum_{j=1}^k \underline{F}_{ij} \cdot \underline{n}_j ds_j \quad (5.11)$$

The indices  $i$  and  $j$  denote the cell and face numbers respectively and  $k$  is the total number of faces of the  $i^{th}$  cell. The right-hand-side of equation (5.11) represents the flux balance for the  $i^{th}$  cell.

The differential form of the equation (5.10) is hyperbolic in time: therefore the flux evaluation is based on the direction of propagation of the information. This is done using upwinding based on the eigensystem of the governing equations.

The numerical flux crossing a cell face is calculated as the exact solution of the approximate Riemann problem ( Roe [45] ). The Riemann problem is defined as the conservation law together with particular initial data consisting of two constant states separated by a single discontinuity ( LeVeque [46] ). The basic principle behind the Roe's approximate Riemann solver is explained below using the one dimensional hyperbolic system of conservation laws,

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = 0 \quad (5.12)$$

with the initial conditions, as given in Figure 5.3.

$$q(x, 0) = \begin{cases} q_L & x < 0 \\ q_R & x > 0 \end{cases} \quad (5.13)$$

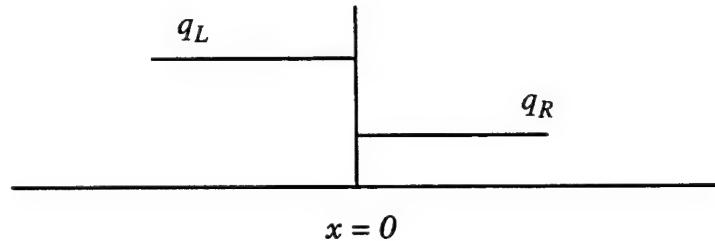


Figure 5.3 Initial Conditions for the Conservation Laws

The above system of conservation laws can be written as

$$\frac{\partial q}{\partial t} + a \frac{\partial q}{\partial x} = 0 \quad (5.14)$$

where  $a = \frac{\partial f}{\partial q}$

Roe's approximate Riemann solver finds the exact solution of the following linear hyperbolic system.

$$\frac{\partial \bar{q}}{\partial t} + \bar{a} \frac{\partial \bar{q}}{\partial x} = 0 \quad (5.15)$$

where  $\bar{a}$  is a locally constant matrix and has to satisfy the following properties (Roe [45]).

- 1) It represents a linear mapping from the vector space  $q$  to the vector space  $f$
- 2) As  $q_L \rightarrow q_R \rightarrow q$ ,  $\bar{a}(q_L, q_R) \rightarrow a(q)$
- 3) For any  $q_L, q_R$ ,  $f(q_R) - f(q_L) = \bar{a}(q_R, q_L) \cdot (q_R - q_L)$
- 4) The eigenvectors of  $\bar{a}$  are linearly independent.

The flux vector at  $x=0$  is then given by

$$f_{x=0} = \frac{1}{2}(f_L + f_R) - \frac{1}{2}|\bar{a}(q_L, q_R)| (q_R - q_L) \quad (5.16)$$

In the case of three dimensional Euler equations the Jacobian matrix  $A$  is defined as  $A = \frac{\partial \mathcal{F}}{\partial Q}$  and the Roe averaged variables appearing in  $\bar{A}$  are evaluated using the following expressions (Roe [45]).

$$\begin{aligned} \bar{\rho} &= \sqrt{\rho_L \rho_R} \\ \bar{v} &= \frac{\sqrt{\rho_L} v_L + v_R \sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} & \bar{u} &= \frac{\sqrt{\rho_L} u_L + u_R \sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \\ \bar{h} &= \frac{\sqrt{\rho_R} h_L + h_R \sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} & \bar{w} &= \frac{\sqrt{\rho_L} w_L + w_R \sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \end{aligned}$$

where  $h$  denotes the total enthalpy per unit volume.

Using the approximate Riemann solver, the flux through a cell face is given by

$$\mathcal{F}_{ij} = \frac{1}{2} [\mathcal{F}(Q_R) + \mathcal{F}(Q_L) - |\bar{A}|(Q_R - Q_L)] \quad (5.17)$$

where  $|\bar{A}| = T|A|T^{-1}$ .  $T$  is a matrix whose columns are the right eigenvectors of  $\bar{A}$ ,  $T^{-1}$  is a matrix with its rows as the left eigenvectors of  $\bar{A}$ , and  $|A|$  is a diagonal matrix whose elements are the absolute values of the eigenvalues of  $\bar{A}$ . The matrix  $\bar{A}$  is evaluated using the Roe averaged variables defined earlier.

The last term of equation (5.17) is separated into three vectors based on the three distinct eigenvalues of  $\bar{A}$ . These vectors are given by

$$|\bar{A}|(Q_R - Q_L) = |\Delta F_1| + |\Delta F_{2,3,4}| + |\Delta F_5| \quad (5.18)$$

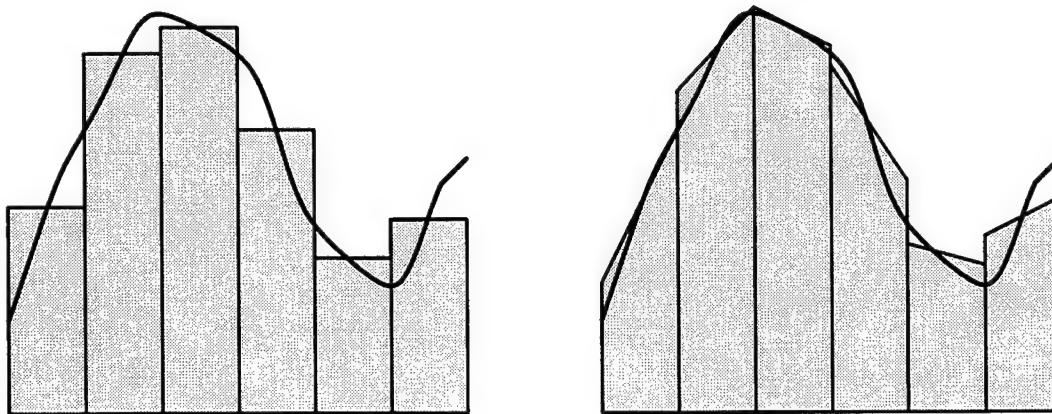
$$\begin{aligned} |\Delta F_1| &= |\bar{U} - \bar{c}| \left( \frac{\Delta p - \bar{\rho} \bar{c} \Delta U}{2 \bar{c}^2} \right) \begin{bmatrix} 1 \\ \bar{u} - \frac{1}{\bar{c}} n_x \\ \bar{v} - \frac{1}{\bar{c}} n_y \\ \bar{w} - \frac{1}{\bar{c}} n_z \\ \bar{H} - \frac{1}{\bar{c}} \bar{U} \end{bmatrix} \\ |\Delta F_{2,3,4}| &= |\bar{U}| \left[ \left( \Delta \rho - \frac{\Delta p}{\bar{c}^2} \right) \begin{bmatrix} \frac{1}{\bar{u}} \\ \frac{1}{\bar{v}} \\ \frac{1}{\bar{w}} \\ \frac{\bar{q}}{2} \end{bmatrix} + \bar{\rho} \begin{bmatrix} 0 \\ \Delta u - n_x \Delta U \\ \Delta v - n_y \Delta U \\ \Delta w - n_z \Delta U \\ \bar{u} \Delta u + \bar{v} \Delta v + \bar{w} \Delta w - \bar{U} \Delta U \end{bmatrix} \right] \\ |\Delta F_5| &= |\bar{U} + \bar{c}| \left( \frac{\Delta p + \bar{\rho} \bar{c} \Delta U}{2 \bar{c}^2} \right) \begin{bmatrix} 1 \\ \bar{u} + \frac{1}{\bar{c}} n_x \\ \bar{v} + \frac{1}{\bar{c}} n_y \\ \bar{w} + \frac{1}{\bar{c}} n_z \\ \bar{H} + \frac{1}{\bar{c}} \bar{U} \end{bmatrix} \end{aligned}$$

where  $U$  is the velocity normal to the face which is given by  $U = u n_x + v n_y + w n_z + n_t$ . The symbol  $\Delta$  represents the jump in the values of the flow variables across the face and is defined as  $\Delta(\cdot) = (\cdot)_R - (\cdot)_L$ . The quantities  $\bar{c}$ , and  $\bar{q}$  are the speed of sound and total velocity respectively, evaluated using the Roe averaged variables.

### 5.3.2 Higher Order Schemes

For a first order accurate scheme, the flow variables are assumed to be constant within each cell. The states of the flow variables on the left and the right sides of a cell face, for the use in the approximate Riemann solver, are taken as the values of the cell averaged data on the respective sides. This procedure avoids the creation of local maxima or minima of the flow variables and preserves monotonicity ( Barth [42] ).

For second order schemes, the flow variables are assumed to be distributed linearly within each cell and the linear distribution is reconstructed from the cell averaged values. Figure 5.4 shows the distribution of the flow variables for first and second order schemes for a one dimensional case. From the figure it is evident that the function can be better represented using linear reconstruction than the piecewise constant representation. This in turn improves the solution accuracy. During the solution process the flow variables at a cell face are extrapolated from the cell averaged values using a linear reconstruction procedure (Barth [42]).



(a) Piecewise Constant Data

(b) Linear Reconstruction

Figure 5.4 Distribution of Flow Variables for First and Second order Scheme.

The piecewise linear reconstruction of the conserved variables from the cell averaged data is done using the Taylor's series expansion. The Taylor's series expansion for a function of multiple variables can be written as,

$$Q(x, y, z) = Q(x_i, y_i, z_i) + \nabla Q(x_i, y_i, z_i) \cdot \Delta \underline{r} + O((\Delta \underline{r})^2) \quad (5.19)$$

where  $\nabla Q$  is the gradient of  $Q$  and  $\Delta \underline{r}$  is the vector from the center of the cell ( $x_i, y_i, z_i$ ) to the desired point ( $x, y, z$ ) and is defined as

$$\Delta \underline{r} = (x - x_i) \underline{i} + (y - y_i) \underline{j} + (z - z_i) \underline{k} \quad (5.20)$$

Using the definition of  $\Delta \underline{r}$  given in equation (5.20), equation (5.19) [neglecting higher order terms] can be written as

$$Q(x, y, z) = Q(x_i, y_i, z_i) + \frac{\partial Q}{\partial x} (x - x_i) + \frac{\partial Q}{\partial y} (y - y_i) + \frac{\partial Q}{\partial z} (z - z_i) \quad (5.21)$$

The gradient of  $Q$  at the cell center is estimated using Green's theorem (Warsi [40]). Green's theorem relates the gradients within a control volume to the surface integral as:

$$\nabla Q = \frac{1}{V_i} \oint_{\partial \Omega} Q \underline{n} ds \quad (5.22)$$

The control volume for integrating equation (5.22) is taken as the volume of the cell itself. This necessitates the estimation of the properties at the nodes. The properties at the nodes are calculated using the weighted average of the properties within the cells surrounding that node. Figure 5.5 shows the cells that are considered for the weighted average of the properties at the node  $N1$ . The weight is taken as the inverse of the distance of the node from the cell center ( Frink et al. [48] ). The resulting equation for the properties at the nodes are written as,

$$Q_{N1} = \frac{\sum_{j=1}^{n_j} \frac{Q_{c_j}}{r_j}}{\sum_{j=1}^{n_j} \frac{1}{r_j}} \quad (5.23)$$

where  $N1$  is the node number,  $c_j$  the neighboring cell number,  $n_j$  the total number of cells surrounding the node  $N1$ , and  $r_j$  the distance of the node from the cell center, i.e.,

$$r_j = \sqrt{(x_c - x_j)^2 + (y_c - y_j)^2 + (z_c - z_j)^2} \quad (5.24)$$

where  $(x_c, y_c, z_c)$  is the cell centroid and  $(x_j, y_j, z_j)$  are the co-ordinates of the node.

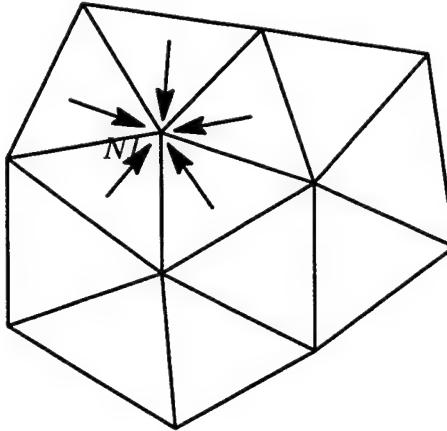


Figure 5.5 Cells Contributing to Node N1 for the Weighted Average

The discretized form of the equation (5.22), which is used for the gradient calculation, can be written in terms of the properties at the nodes as,

$$(\nabla Q)_i = \frac{1}{V_i} \sum_{k=1}^{n_k} Q_f n_{ik} ds_k \quad (5.25)$$

where  $n_k$  is the number of cell faces for the cell  $i$  and  $Q_f$  is the average value of the flow variables at the nodes which forms the cell face  $k$ .

The derivative of the conserved variables in the three co-ordinate directions can be explicitly written as (from equation (5.25)),

$$\frac{\partial Q_i}{\partial x} = \frac{1}{V_i} \sum_{k=1}^{n_k} Q_f(n_x)_k \quad (5.26)$$

$$\frac{\partial Q_i}{\partial y} = \frac{1}{V_i} \sum_{k=1}^{n_k} Q_f(n_y)_k \quad (5.27)$$

$$\frac{\partial Q_i}{\partial z} = \frac{1}{V_i} \sum_{k=1}^{n_k} Q_f(n_z)_k \quad (5.28)$$

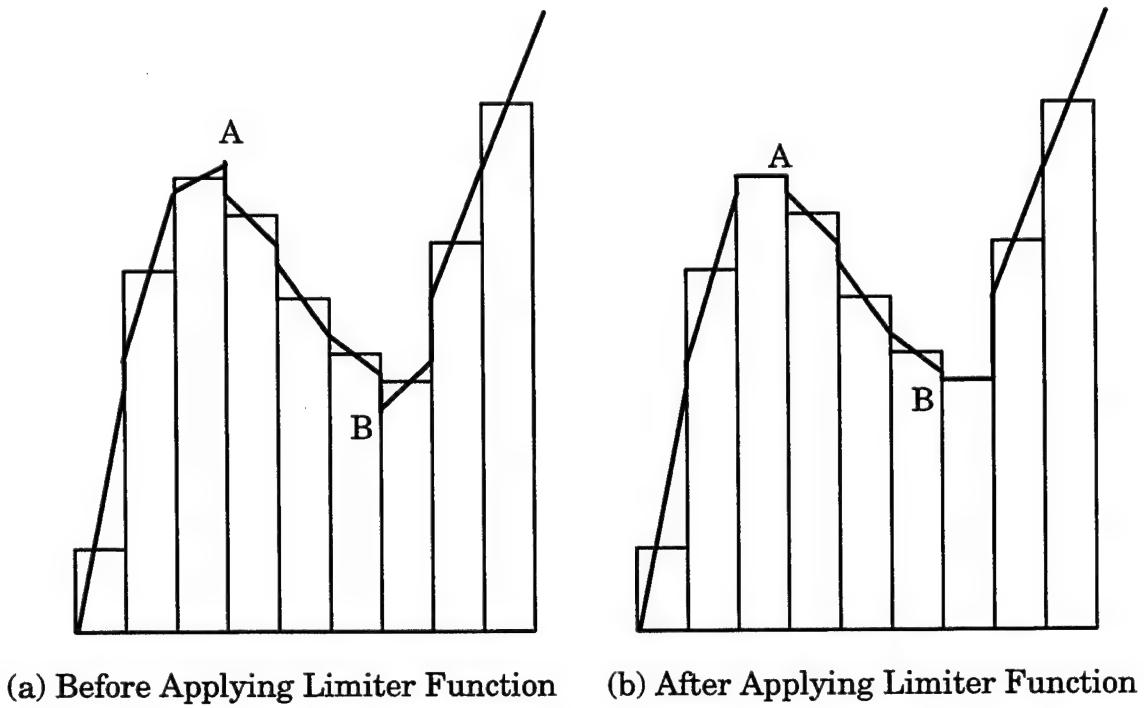
Using equations (5.26)–(5.28) in equation (5.21), the linear distribution of the properties with a cell can be calculated.

One of the disadvantage of this approach is that the information from the wrong side of the cell face also contribute to the weighted averaged values of the conserved variables at the nodes. This can be overcome by using upwind biased gradients as discussed by Cabello [49]. The higher order calculations based on the upwind biased gradients are not implemented in the present study because of the higher memory overheads required to store the information about the cells used for the upwind biased gradient calculation and the coding complexity.

### 5.3.3 Limiter Functions

For cell centered schemes, the cell averaged values are stored at the cell center. The gradient within a cell is calculated based on the values in the neighboring cells. Using the gradients and the cell averaged values, the linear distribution is reconstructed as shown in Figure 5.6. From Figure 5.6(a), one can see that local extrema are created at locations *A* and *B*. At location *A* the reconstructed value is more than the cell averaged values of the neighboring cells. Similarly, at point *B* the reconstructed value of the variable is lower than that of the

neighboring cells. These non-physical values will cause spurious oscillations of the flow variables near regions of high solution gradient. This needs to be avoided to preserve monotonicity and to avoid the development of spurious oscillations near discontinuities. This is achieved by the application of a limiter function ( Barth [42] ). However it should be noted that the limiter function reduces the order of accuracy in those regions.



**Figure 5.6 Effect of Limiter Function on Linear Reconstruction**

The limiter function is applied to the Taylor's series expansion to avoid the creation of local minima and maxima. The Taylor's series expansion, given in equation (5.19), is modified with the limiter function as,

$$Q(x, y, z) = Q(x_i, y_i, z_i) + \phi_i \nabla Q(x_i, y_i, z_i) \cdot \Delta r + O((\Delta r)^2) \quad (5.29)$$

The limiting function  $\phi_i$  is constructed to satisfy the monotonicity principle and has to satisfy the following condition.

If  $Q_i^{min} = \min(Q_{Ci}, Q_{adj})$

$$Q_i^{max} = \max(Q_{Ci}, Q_{adj})$$

$$\text{then } Q_i^{min} \leq Q(x,y) \leq Q_i^{max}$$

$C_i$  represents the cell for which the limiter function is being evaluated and the subscript  $adj$  represents the cells adjoining to  $C_i$ . Two different limiter functions were considered and their effect on the convergence of the numerical scheme were studied. The two limiter are discussed below.

### 5.3.3.1 Barth's Limiter

A limiter function was developed by Barth [42] to satisfy the above mentioned properties and is defined as

$$\phi_{ij} = \begin{cases} \min\left(1, \frac{Q_i^{max} - Q_{ij}}{Q_i - Q_{ij}}\right) & \text{if } Q_i - Q_{ij} > 0 \\ \min\left(1, \frac{Q_i^{min} - Q_{ij}}{Q_i - Q_{ij}}\right) & \text{if } Q_i - Q_{ij} < 0 \\ 1 & \text{if } Q_i - Q_{ij} = 0 \end{cases} \quad (5.30)$$

$$\text{and } \phi_i = \min(\phi_{ij}) \quad (5.31)$$

In the above relations,  $Q_{ij}$  is calculated using equation (5.19).

### 5.3.3.2 Venkatakrishnan's Limiter

Even though Barth's limiter gives accurate results for the flow solutions, the convergence stalls after a few orders of magnitude drop in the residual. To avoid this, Venkatakrishnan [50] proposed another limiter for unstructured grids. This limiter is basically an extension of the van Albada limiter, developed for structured grids. The new limiter is defined as

$$\phi_{ij} = \frac{1}{\Delta_-} \left[ \frac{(\Delta_+^2 + \epsilon^2)\Delta_- + 2\Delta_-^2\Delta_+}{\Delta_+^2 + 2\Delta_-^2 + \Delta_-\Delta_+ + \epsilon^2} \right] \quad (5.32)$$

where  $\varepsilon^2$  is taken as  $(K \Delta x)^3$  and  $\Delta x$  represents the average edge size of triangles, i.e., if the physical domain is covered with same number of equilateral triangles of equal size, then  $\Delta x$  is the edge size of those triangles and  $K$  is a threshold parameter. When  $K=0.0$ , the limiter will be active everywhere in the field, whereas a very high value of  $K$  implies effectively no limiting. Typically  $K$  is set to 0.1 to 0.3.

The other variables appearing in equation (5.32) are defined as follows,

$$\left. \begin{array}{l} \Delta_- = Q_{ij} - Q_i \\ \Delta_+ = Q_i^{\max} - Q_i \end{array} \right\} \quad \text{if } Q_{ij} - Q_i > 0 \quad (5.33)$$

$$\left. \begin{array}{l} \Delta_- = Q_{ij} - Q_i \\ \Delta_+ = Q_i^{\min} - Q_i \end{array} \right\} \quad \text{if } Q_{ij} - Q_i < 0 \quad (5.34)$$

In order to avoid division by a small value of  $\Delta_-$  in equation NO TAG it is replaced by  $SIGN(\Delta_-)(|\Delta_-| + \omega)$  and  $\omega$  is set to  $10^{-12}$ , where  $SIGN(x)$  is the sign of  $x$ . Results are presented for the effect of these limiters on convergence in Section 6.2.

One of the drawbacks of the Roe scheme is that it allows the creation of nonphysical expansion shocks in some cases. Another drawback ( more likely at higher Mach numbers) is that it admits spurious solutions along the stagnation line ahead of bow shocks, which is known as the carbuncle phenomenon ( Quirck [47] ). In order to avoid these phenomena, the wave speeds are modified when the absolute value of the eigenvalues of  $\bar{A}$  are less than 0.2. This correction to the eigenvalues are applied when the absolute value of the eigenvalues are in the neighborhood of zero. The correction is started from 0.2 as a compromise between the numerical stability and the accuracy. The modification is defined as a quadratic function, i.e.,

$$|\lambda'| = A |\lambda|^2 + B \quad (5.35)$$

with the constraints

$$\lambda' = 0.2 \quad \text{when} \quad \lambda = 0.2$$

$$\left( \frac{d\lambda'}{d\lambda} \right)_{\lambda=0.2} = 1 \quad \left( \frac{d\lambda'}{d\lambda} \right)_{\lambda=0} = 0$$

This results in  $A = 2.5$  and  $B = 0.1$ .

#### 5.4 Viscous Flux Calculations

The viscous fluxes are treated as source terms and are evaluated explicitly. The viscous flux terms, that involve the components of the viscous stress tensor, are defined in Chapter II. The viscous stress tensor in turn is a function of velocity and temperature gradients.

The gradients of velocity and temperature are estimated by the application of Green's theorem to a control volume. In this case the control volume is taken as the cell itself. The properties at the nodes are computed using the weighted averaging procedure explained in Section 5.3.2.

The viscous stresses at the cell face are taken as the average of those in the cells on both sides of the face. The averaging is a good approximation for the viscous stresses, since they are diffusive in nature. This gives a second order approximation but the truncation error, for the structured part, is proportional to four time the square of the grid spacing. The contribution of the viscous fluxes to the net flux crossing the control surface are estimated as shown for the convective terms given in Section 5.3.1.

#### 5.5 Time Discretization

The fully discretized form of equation (5.10) can be written as,

$$V_i \frac{Q_i^{n+1} - Q_i^n}{\Delta t} = - \sum_{j=1}^k \underline{F}_{ij} \cdot \underline{n}_j ds_j \quad (5.36)$$

The time level at which the right hand side of equation (5.36) is computed results in different numerical schemes. The evaluation of the right hand side of equation (5.36) at  $n^{th}$  time level results in an explicit scheme and at  $(n+1)^{th}$  time level will result in an implicit scheme. For the explicit scheme the time step is restricted by the stability criteria and for the implicit scheme the flux vector has to be linearized before the calculation of the flux crossing the cell faces. This results in set of linear equations that have to be solved to obtain the conserved variables at the  $(n+1)^{th}$  time level. The detailed description of these methods are given in the following subsections.

### 5.5.1 Explicit Scheme

For explicit time integration, a four stage Runge–Kutta method is employed (Jameson et al. [51]). The flow variables at  $(n+1)^{th}$  time step are obtained from the variables at the  $n^{th}$  time step in four stages. The time integration can be written as

$$\begin{aligned}
 Q^{(0)} &= Q^{(N)} \\
 Q^{(1)} &= Q^{(0)} + \alpha_1 \frac{\Delta t}{V_i} R(Q^{(0)}) \\
 Q^{(2)} &= Q^{(0)} + \alpha_2 \frac{\Delta t}{V_i} R(Q^{(1)}) \\
 Q^{(3)} &= Q^{(0)} + \alpha_3 \frac{\Delta t}{V_i} R(Q^{(2)}) \\
 Q^{(4)} &= Q^{(0)} + \alpha_4 \frac{\Delta t}{V_i} R(Q^{(3)}) \\
 Q^{(N+1)} &= Q^{(4)}
 \end{aligned} \tag{5.37}$$

where  $R(Q)$  is the right hand side of equation (3.29) and the coefficients used are  $\alpha_1 = 0.15$ ,  $\alpha_2 = 0.3275$ ,  $\alpha_3 = 0.57$  and  $\alpha_4 = 1.0$ . These weighting coefficients are available in the literature and have been experimentally determined for structured upwind codes (Turkel and Van Leer [53]). For steady state problems faster

convergence is achieved using local time stepping in which the maximum permissible time step is used for each individual cells.

### 5.5.2 Implicit Scheme

In the case of implicit schemes the numerical flux crossing the cell face is a function of the conserved variables at the  $(n+1)^{th}$  time step (Koomullil [52]). For the moving grids the cell volume is changing at every time step and has to satisfy the geometric conservation law. The detailed description about the problems associated with the moving body problems are given in Chapter V. The discretized equations then become

$$V_i \frac{\Delta Q_i^n}{\Delta t} = - \sum_{j=1}^k F_{ij}^{n+1} \cdot n_j ds_j = R_i^{n+1} \quad (5.38)$$

A linearization of the RHS about the  $n^{th}$  time level results in

$$R_i^{n+1} = R_i^n + \left( \frac{\partial R_i}{\partial Q} \right)^n \Delta Q^n \quad (5.39)$$

Substituting equation (5.39) into equation (5.38), yields

$$\frac{V_i}{\Delta t} I \Delta Q_i^n - \left( \frac{\partial R_i}{\partial Q} \right)^n \Delta Q^n = R_i^n \quad (5.40)$$

Using the the summation convention over the edges the above system can be written as,

$$\frac{V_i}{\Delta t} I \Delta Q_i^n + \sum_{j=1}^k \left( \frac{\partial \mathcal{F}_{ij}}{\partial Q} \right)^n \Delta Q^n = R_i^n \quad (5.41)$$

Utilizing the fact that  $\mathcal{F}_{ij}$  is a function of conserved variables on left and right sides of each face, the above system of equations is written as,

$$\frac{V_i}{\Delta t} I \Delta Q_i^n + \sum_{j=1}^k \left[ \left( \frac{\partial \mathcal{F}_{ij}}{\partial Q_L} \right)^n \Delta Q_L^n + \left( \frac{\partial \mathcal{F}_{ij}}{\partial Q_R} \right)^n \Delta Q_R^n \right] = R_i^n \quad (5.42)$$

According to the sign convention of the surface normal, the surface normal points from left cell to the right cell, the cell number on the left hand side  $L$  and the cell number  $i$  are the same. Then the above equation reduces to,

$$\left[ \frac{V_i}{\Delta t} I + \sum_{j=1}^k \left( \frac{\partial \mathcal{F}_{ij}}{\partial Q_i} \right)^n \right] \Delta Q_i^n + \sum_{j=1}^k \left[ \left( \frac{\partial \mathcal{F}_{ij}}{\partial Q_R} \right)^n \Delta Q_R^n \right] = R_i^n \quad (5.43)$$

The Jacobian matrices  $\frac{\partial \mathcal{F}_{ij}}{\partial Q_i}$  and  $\frac{\partial \mathcal{F}_{ij}}{\partial Q_R}$  can be estimated either by an approximate analytic method or by a numerical approach. These are discussed in detail in Sections 3.4.2.2 and 3.4.2.3.

## 5.6 Newton Iterations

For unsteady calculations, it is important to drive the unsteady residual to zero for better resolution of the physical phenomenon. This can be done by using Newton iterations. The basic principle behind Newton's method (Whitfield and Taylor [54], Whitfield [56]) and its implementation are given below.

The implicit form of the discretized governing equations are written as

$$\mathcal{H} \equiv V_i \left( \frac{Q_i^{n+1} - Q_i^n}{\Delta t} \right) + \sum_{j=1}^k \mathcal{F}_{ij}(Q^{n+1}) = 0 \quad (5.44)$$

The conserved variable vector  $Q^{n+1}$  has to satisfy equation (5.44) to achieve a time accurate solution. This system can be written as  $\mathcal{H}(Q^{n+1}) = 0$ . Newton's method for this system is given by

$$\mathcal{H}'(Q^{n+1,m})(Q^{n+1,m+1} - Q^{n+1,m}) = -\mathcal{H}(Q^{n+1,m}) \quad (5.45)$$

The prime denotes the derivative with respect to the conserved variable vector  $Q$ .

The above equation can be expanded as

$$\frac{\partial \mathcal{H}}{\partial Q_i} \Delta Q_i^{n+1,m} + \frac{\partial \mathcal{H}}{\partial Q_j} \Delta Q_j^{n+1,m} = -\mathcal{H}(Q_i^{n+1,m}) \quad (5.46)$$

$$\text{where } \Delta Q^{n+1,m} = Q^{n+1,m+1} - Q^{n+1,m} \quad (5.47)$$

where  $j$  represents the cells the surrounds the cell  $i$ . The superscript  $m$  in equations (5.45)–(5.47) represents the index for Newton iterations. Using equation (5.44), equation (5.46) is written as,

$$\left[ \frac{V_i}{\Delta t} I + \sum_{j=1}^k \left( \frac{\partial \mathcal{F}_{ij}}{\partial Q_i} \right)^{n+1,m} \right] \Delta Q_i^{n+1,m} + \sum_{j=1}^k \left( \frac{\partial \mathcal{F}_{ij}}{\partial Q_j} \right)^{n+1,m} \Delta Q_j^{n+1,m} = \\ - \left[ V_i \left( \frac{Q_i^{n+1,m} - Q_i^n}{\Delta t} \right) + \sum_{j=1}^k \mathcal{F}_{ij}(Q^{n+1,m}) \right] \quad (5.48)$$

At the start of Newton iterations,  $Q^{n+1,0}$  is set as  $Q^n$  and the linear system will reduce to

$$\left[ \frac{V_i}{\Delta t} I + \sum_{j=1}^k \left( \frac{\partial \mathcal{F}_{ij}}{\partial Q_i} \right)^n \right] \Delta Q_i^n + \sum_{j=1}^k \left( \frac{\partial \mathcal{F}_{ij}}{\partial Q_j} \right)^n \Delta Q_j^n = - \sum_{j=1}^k \mathcal{F}_{ij}(Q^n) \quad (5.49)$$

This is the same as the implicit scheme which is used for the steady state calculations as given in equation (5.43). As more Newton iterations are used, then  $\Delta Q^{n+1,m}$  approaches zero resulting in time accurate solutions. For more details about Newton iterations (Whitfield and Taylor [54]).

## 5.7 Approximate Analytic Jacobians

The flux crossing the cell face is written as,

$$\mathcal{F}_{ij} = \frac{1}{2}(\mathcal{F}(Q_R) + \mathcal{F}(Q_L) - |\bar{A}|(Q_R - Q_L)) \quad (5.50)$$

The Roe averaged matrix is  $|\bar{A}|$  is a nonlinear function of the conserved variables at left and right sides of the cell face. For simplicity the approximate analytic Jacobians are calculated assuming the Roe averaged matrix  $|\bar{A}|$  as constant (Whitfield [55]). The above equation can be differentiated analytically with respect to  $Q_R$  and  $Q_L$  resulting

$$\frac{\partial \mathcal{F}_{ij}}{\partial Q_R} = \frac{1}{2} \left( \frac{\partial \mathcal{F}(Q_R)}{\partial Q_R} + |\bar{A}| \right) = \frac{1}{2} (A_R + |\bar{A}|) \quad (5.51)$$

$$\frac{\partial \mathcal{F}_{ij}}{\partial Q_L} = \frac{1}{2} \left( \frac{\partial \mathcal{F}(Q_L)}{\partial Q_L} - |\bar{A}| \right) = \frac{1}{2} (A_L - |\bar{A}|) \quad (5.52)$$

Substitution of the above expressions into equation (5.43) will result in a block sparse matrix.

### 5.8 Numerical Jacobians

The Jacobians can be calculated from first principles. Each element in the Jacobian can written as  $a_{ij} = \frac{\partial \mathcal{F}_i(Q)}{\partial Q_j}$ . The first order form of these quantities can be evaluated using a finite difference formula (Whitfield [55], Vanden [57]).

$$a_{ij} = \frac{\mathcal{F}_i(Q + he_j) - \mathcal{F}_i(Q)}{h} \quad (5.53)$$

where  $e_j$  is the  $j^{th}$  unit vector and  $h$  is taken as the square root of machine zero. The choice of approximate analytical or numerical Jacobians has a strong influence on convergence to steady state

### 5.9 Sparse Matrix System

Equation (5.43) will result in a linear sparse block matrix system of the form  $A x = b$ . The number of non-zero entries in each row of  $A$  depends on the number of neighbors of the cell, whose cell number is the row under consideration. If the cell  $i$  has  $p$  neighbors, then the  $i^{th}$  row of  $A$  will have  $p+1$  non-zero elements. The matrix structure with a sample grid is explained in detail in NO TAG. The symbol  $\bullet$  in the matrix represents a non-zero entry. Each non-zero entry in this matrix will be either a  $4 \times 4$  block for two dimensional cases or  $5 \times 5$  for three dimensional cases. The resulting sparse matrix is solved using the Generalized Minimal RESidual (GMRES) ( Saad and Schultz [58] ) method. An incomplete LU decomposition is used as the preconditioner.

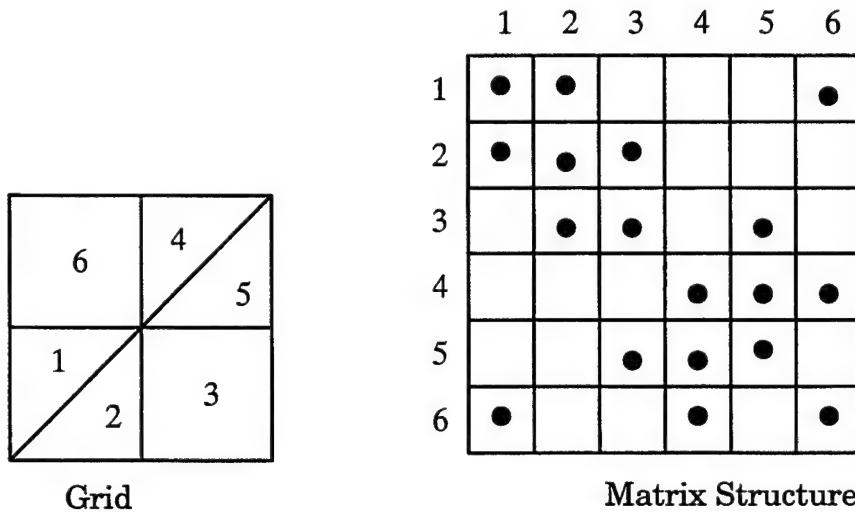


Figure 5.7 Sparse Matrix Structure Resulting from Implicit Scheme

### 5.10 Time Step Calculation

The time steps used for the computations of three dimensional problems are based on a direct extension of the stability analysis of two dimensional cases. The time step is calculated using the relation ( Frink, et al. [48] )

$$\Delta t_i \leq CFL \frac{V_i}{A_i + B_i + C_i} \quad (5.54)$$

where  $A_i = (|u_i| + c_i) S_i^{(x)}$

$$B_i = (|v_i| + c_i) S_i^{(y)}$$

$$C_i = (|w_i| + c_i) S_i^{(z)}$$

$V_i$  is the cell volume,  $c_i$  the local speed of sound, and  $S_i^{(x)}$ ,  $S_i^{(y)}$ , and  $S_i^{(z)}$  are the projected areas of the cell  $i$  along  $x$ ,  $y$ , and  $z$  coordinate directions respectively.

### 5.11 Boundary Conditions

The boundary conditions are derived from the sign of the eigenvalues and the principle of wave propagation (Whitfield and Janus [60]). Local one dimensional flow is assumed for these derivations. The boundary conditions for different flow situations are discussed below.

#### 5.11.1 Supersonic Inflow or Outflow

All the eigenvalues have the same sign for the supersonic inflow or outflow conditions. Therefore, flow variables are specified for the inflow condition and are extrapolated for the outflow conditions.

#### 5.11.2 Subsonic Inflow

In this case four eigenvalues are of the same sign and the fifth one is of the opposite sign. According to the sign convention the direction of the unit normal vector to the control surface is always pointing outward from the control surface. Therefore the subsonic inflow will always be in a direction opposite to the unit normal. This will result in the following conditions at the boundary.

$$p_b = \frac{1}{2} (p_\infty + p_L - \rho_0 c_0 [\tilde{n}_x (u_\infty - u_L) + \tilde{n}_y (v_\infty - v_L) + \tilde{n}_z (w_\infty - w_L)])$$

$$\rho_b = \rho_\infty + \frac{p_b - p_\infty}{c_0^2}$$

$$u_b = u_\infty - \tilde{n}_x \frac{p_\infty - p_b}{\rho_0 c_0}$$

$$v_b = v_\infty - \tilde{n}_y \frac{p_\infty - p_b}{\rho_0 c_0}$$

$$w_b = w_\infty - \tilde{n}_z \frac{p_\infty - p_b}{\rho_0 c_0}$$

where the subscript  $\infty$  refers to the freestream conditions,  $L$  refers to the cell inside the domain and  $b$  is the point on the computational boundary.

### 5.11.3 Subsonic Outflow

Similar to the subsonic inflow, in this case also four eigenvalues are of the same sign and the fifth one is of the opposite sign. Therefore one characteristic variable is specified in the form of pressure and the other four are determined from the information within the domain. So the boundary conditions are determined as follows

$$p_b = p_{specified}$$

$$\rho_b = \rho_L + \frac{p_b - p_L}{c_0^2}$$

$$u_b = u_L + \tilde{n}_x \frac{p_L - p_b}{\rho_0 c_0}$$

$$v_b = v_L + \tilde{n}_y \frac{p_L - p_b}{\rho_0 c_0}$$

$$w_b = w_L + \tilde{n}_z \frac{p_L - p_b}{\rho_0 c_0}.$$

#### 5.11.4 Impermeable Wall

For the impermeable wall, there is no flow across it and so the first three eigenvalues are zero. The fourth and fifth eigenvalues have opposite signs. These will result in the following relations for the boundary values.

$$p_b = p_L + \rho_0 c_0 (\tilde{n}_x u_L + \tilde{n}_y v_L + \tilde{n}_z w_L + \tilde{n}_t)$$

$$\rho_b = \rho_L + \frac{p_b - p_L}{c_0^2}$$

$$u_b = u_L - \tilde{n}_x (\tilde{n}_x u_L + \tilde{n}_y v_L + \tilde{n}_z w_L + \tilde{n}_t)$$

$$v_b = v_L - \tilde{n}_y (\tilde{n}_x u_L + \tilde{n}_y v_L + \tilde{n}_z w_L + \tilde{n}_t)$$

$$w_b = w_L - \tilde{n}_z (\tilde{n}_x u_L + \tilde{n}_y v_L + \tilde{n}_z w_L + \tilde{n}_t)$$

where  $L$  refers to the first cell from the boundary. For the viscous calculations the velocity components on the body are set to the local velocity of the body and the pressure and density are calculated in the same manner as that of inviscid case.

#### 5.12 Turbulence Modeling

Viscosity can be viewed as consisting of two contributions: laminar and turbulent. The laminar viscosity, which is a fluid property, is usually a function of temperature and can be estimated using Sutherland's formula ( Anderson et al. [41]). The turbulent viscosity, on the other hand, is a function of the flow and needs to be evaluated using some model. In this study the turbulent viscosity is estimated using the Spalart–Allmaras one equation model (Spalart and Allmaras [35]) and the Reynolds stress is modelled using the Boussinesq hypothesis ( Warsi [40] ), i.e., it is written in terms of the turbulent viscosity and the gradients of the flow variables. Spalart–Allmaras turbulence model is a pointwise model, which makes the application of this model for unstructured or hybrid grids easier. This model solves a second order partial differential equation for the variable  $\bar{v}$  and the turbulent kinematic viscosity  $\nu_t$  is estimated from  $\bar{v}$  by multiplying by a damping function  $f_{vJ}$ .

After non-dimensionalization, the Spalart–Allmaras one equation turbulence model in vector invariant form can be written as,

$$\begin{aligned}\frac{\partial \bar{v}}{\partial t} = & - \underline{V} \cdot \nabla \bar{v} + C_{b1} \tilde{S} \bar{v} + \frac{1}{\sigma \text{Re}_L} (\nabla \cdot ((v + \bar{v}) \nabla \bar{v})) \\ & + \frac{C_{b2}}{\sigma \text{Re}_L} (\nabla \bar{v})^2 - \frac{C_{\omega 1} f_{\omega}}{\text{Re}_L} \left( \frac{\bar{v}}{d} \right)^2\end{aligned}\quad (5.55)$$

Correspondingly the variables appearing in the above equations will change to,

$$\chi = \frac{\bar{v}}{v} ;$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3} ; \quad v_t = \bar{v} f_{v1}$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} ; \quad \tilde{S} = S + \frac{1}{\text{Re}_L} \left( \frac{\bar{v}}{\kappa^2 d^2} \right) f_{v2}$$

$$f_{\omega} = g \left( \frac{1 + C_{\omega 3}^6}{g^6 + C_{\omega 3}^6} \right)^{\frac{1}{6}} ; \quad g = r + C_{\omega 2} (r^6 - r) ;$$

$$r = \frac{1}{\text{Re}_L} \left( \frac{\bar{v}}{\tilde{S} \kappa^2 d^2} \right)$$

$$C_{b1} = 0.1355 ; \quad \sigma = 2/3 ; \quad C_{b2} = 0.622 ; \quad \kappa = 0.41$$

$$C_{\omega 1} = \frac{C_{b1}}{\kappa^2} + \frac{1 + C_{b2}}{\sigma} ; \quad C_{\omega 2} = 0.3 ; \quad C_{\omega 3} = 2.0 ; \quad C_{v1} = 7.1$$

The different terms appearing in equation (5.55) can be divided into convective, production, diffusion, and destruction terms. So the transport equation for the turbulent viscosity parameter is written as,

Time rate of change of viscosity parameter = –Convection + Production +  
Diffusion – Destruction.

where the convection is given by  $\underline{V} \cdot \nabla \bar{\nu}$ . Production of the turbulent viscosity is due to vorticity and is given by  $C_{b1} \tilde{S} \bar{\nu}$ . The diffusion term is a function of the gradient of the viscosity parameter and is defined as

$$\frac{1}{\text{Re}_L \sigma} (\nabla \cdot ((\nu + \bar{\nu}) \nabla \bar{\nu})) + \frac{C_{b2}}{\text{Re}_L \sigma} (\nabla \bar{\nu})^2$$

Neglecting the gradient of laminar viscosity, the diffusion terms can be written as,

$$\frac{1 + C_{b2}}{\text{Re}_L \sigma} (\nabla \cdot ((\nu + \bar{\nu}) \nabla \bar{\nu})) - \frac{C_{b2}}{\text{Re}_L \sigma} (\nu + \bar{\nu}) \nabla^2 \bar{\nu}$$

The destruction term for the turbulent viscosity parameter is taken as inversely proportional to the square of the distance to the solid wall and is given by,

$$\frac{C_{\omega 1} f_{\omega}}{\text{Re}_L} \left( \frac{\bar{\nu}}{d} \right)^2$$

### 5.12.1 Integral Formulation of the Turbulence Model

Equation (5.55) with the simplified diffusion terms can be written as,

$$\begin{aligned} \frac{\partial \bar{\nu}}{\partial t} = & - \underline{V} \cdot \nabla \bar{\nu} + C_{b1} \tilde{S} \bar{\nu} + \frac{(1 + c_{b2})}{\text{Re}_L \sigma} (\nabla \cdot ((\nu + \bar{\nu}) \nabla \bar{\nu})) \\ & - \frac{C_{b2}}{\text{Re}_L \sigma} (\nu + \bar{\nu}) \nabla^2 \bar{\nu} - \frac{C_{\omega 1} f_{\omega}}{\text{Re}_L} \left( \frac{\bar{\nu}}{d} \right)^2 \end{aligned} \quad (5.56)$$

Integrating equation (5.56) over a control volume yields

$$\begin{aligned}
\int_{\Omega} \frac{\partial \bar{v}}{\partial t} d\Omega = & - \int_{\Omega} \underline{V} \cdot \nabla \bar{v} d\Omega + \int_{\Omega} C_{b1} \tilde{S} \bar{v} d\Omega \\
& + \frac{(1 + C_2)}{\text{Re}_L \sigma} \int_{\Omega} \nabla \cdot ((\nu + \bar{v}) \nabla \bar{v}) d\Omega \\
& - \frac{C_{b2}}{\text{Re}_L \sigma} \int_{\Omega} (\nu + \bar{v}) \nabla^2 \bar{v} d\Omega \\
& - \frac{C_{\omega 1}}{\text{Re}_L} \int_{\Omega} f_{\omega} \left( \frac{\bar{v}}{d} \right)^2 d\Omega
\end{aligned} \tag{5.57}$$

Using the divergence theorem for the surface integral, equation (5.57) can be transformed to,

$$\begin{aligned}
\int_{\Omega} \frac{\partial \bar{v}}{\partial t} d\Omega = & - \int_{\Omega} \underline{V} \cdot \nabla \bar{v} d\Omega + \int_{\Omega} C_{b1} \tilde{S} \bar{v} d\Omega \\
& + \frac{(1 + C_2)}{\text{Re}_L \sigma} \oint_{\partial\Omega} (\nu + \bar{v}) \nabla \bar{v} \cdot \underline{n} ds \\
& - \frac{C_{b2}}{\text{Re}_L \sigma} \int_{\Omega} (\nu + \bar{v}) \nabla^2 \bar{v} d\Omega \\
& - \frac{C_{\omega 1}}{\text{Re}_L} \int_{\Omega} f_{\omega} \left( \frac{\bar{v}}{d} \right)^2 d\Omega
\end{aligned} \tag{5.58}$$

### 5.12.2 Numerical Procedure

As in the case of Navier–Stokes equations, equation (5.58) is also solved using a cell centered scheme. The values of  $\bar{v}$ , stored at the cell center, are assumed to be cell averaged values. The discretized form of equation (5.58) is

$$\begin{aligned}
\frac{\bar{v}^{n+1} - \bar{v}^n}{\Delta t_i} V_i = & - \sum_{j=1}^k (U^+ \bar{v}_i + U^- \bar{v}_{n(j)}) ds_j + C_{b1} (\tilde{S} \bar{v})_i V_i \\
& + \frac{(1 + C_{b2})}{\sigma \text{Re}_L} \sum_{j=1}^k (\nu + \bar{v})_{ij} (\nabla \bar{v} \cdot \underline{n})_{ij} ds_j
\end{aligned}$$

$$\begin{aligned}
& - \frac{C_{b2}}{\sigma \text{Re}_L} (\nu + \bar{\nu})_i \nabla^2 \bar{\nu}_i V_i \\
& - \frac{C_{\omega 1}}{\text{Re}_L} \left( f_\omega \left( \frac{\bar{\nu}}{d} \right)^2 \right)_i V_i
\end{aligned} \tag{5.59}$$

where  $k$  is the number of edges of cell  $i$ ,  $\underline{n}$  is the unit normal to the face,  $ds_j$  is the length of the edge  $j$ ,  $n_{(j)}$  is the cell that shares the  $j^{th}$  edge of the cell  $i$ , and  $V_i$  is the cell volume. The variables  $U^+$  and  $U^-$  are defined as

$$\begin{aligned}
U^+ &= \frac{1}{2} (U + |U|) \\
U^- &= \frac{1}{2} (U - |U|)
\end{aligned}$$

where  $U$  is the contravariant velocity. The laminar and turbulent viscosity at the cell faces are taken as the average values of those on either side of the cell face.

The gradient of the turbulent viscosity parameter  $\bar{\nu}$  appear in equation (5.59) in the surface integral. So the gradient is calculated on the cell face. The gradient at the cell faces are calculated using Green's theorem and a weighted averaging procedure. The value of  $\bar{\nu}$  at the nodes are calculated by a weighted average of  $\bar{\nu}$  within the cells surrounding the node. The weight is taken as inverse of the distance between the cell center and the node.

$$\nabla \bar{\nu} = \frac{1}{V_i} \oint_{\partial\Omega} \bar{\nu} \underline{n} ds \tag{5.60}$$

The gradients at the cell face are calculated based on a control volume surrounding that face. The control volume is taken as the volume enclosed by the nodes connecting the face and the cell centers as shown in Figure 5.8

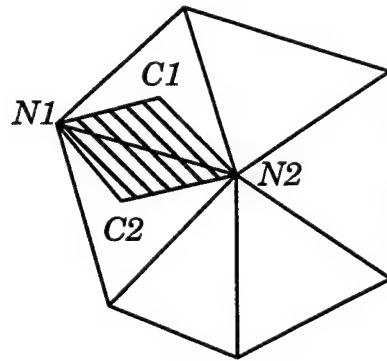


Figure 5.8 Control Volume for Edge Connecting Nodes  $N1$  and  $N2$

### 5.12.3 Explicit Method

For the explicit solution of the equation (5.59), a four stage Runge–Kutta method is utilized. Denoting the right hand side of equation (5.59) as  $R(\bar{v})$ , the four stage Runge–Kutta method can be written as,

$$\bar{v}^{(0)} = \bar{v}^n$$

$$\bar{v}^{(1)} = \bar{v}^n + \alpha_1 \Delta t R(\bar{v}^{(0)})$$

$$\bar{v}^{(2)} = \bar{v}^n + \alpha_2 \Delta t R(\bar{v}^{(1)})$$

$$\bar{v}^{(3)} = \bar{v}^n + \alpha_3 \Delta t R(\bar{v}^{(2)})$$

$$\bar{v}^{(4)} = \bar{v}^n + \alpha_4 \Delta t R(\bar{v}^{(3)})$$

$$\bar{v}^{(n+1)} = \bar{v}^{(4)}$$

where  $\alpha_1 = 0.0833$   $\alpha_2 = 0.2069$   $\alpha_3 = 0.4265$   $\alpha_4 = 1.0$

### 5.12.4 Implicit Method

For the implicit scheme the turbulent viscosity parameter  $\bar{v}$  appearing in the right hand side of Equation (5.59) is taken at  $(n+1)^{th}$  time level. Equation (5.59) can be written as,

$$\frac{V_i (\Delta \bar{v})^n}{\Delta t_i} = C(\bar{v}^{n+1}) + P(\bar{v}^{n+1}) + D_1(\bar{v}^{n+1}) + D_2(\bar{v}^{n+1}) \quad (5.61)$$

where  $C$ ,  $P$ ,  $D_1$ , and  $D_2$  are convective, production, diffusion, and destruction terms respectively. Linearization of the convective term yields,

$$C(\bar{v}^{n+1}) = C(\bar{v}^n) + \frac{\partial C}{\partial \bar{v}_i} (\Delta \bar{v}_i)^n + \frac{\partial C}{\partial \bar{v}_{n(j)}} (\Delta \bar{v}_{n(j)})^n \quad (5.62)$$

The Jacobians appearing in the Equation (5.62) can be calculated analytically and are given by,

$$\frac{\partial C}{\partial \bar{v}_i} = U^+ \quad (5.63)$$

$$\frac{\partial C}{\partial \bar{v}_{n(j)}} = U^- \quad (5.64)$$

Similarly linearizing the production term, one has,

$$P(\bar{v}^{n+1}) = P(\bar{v}^n) + \frac{\partial P}{\partial \bar{v}_i} (\Delta \bar{v}_i)^n \quad (5.65)$$

The production is not an explicit function of  $\bar{v}$ , and so the Jacobian is evaluated numerically as given below:

$$\frac{\partial P}{\partial \bar{v}} = \frac{P(\bar{v} + h) - P(\bar{v})}{h} \quad (5.66)$$

where  $h$  is the square root of machine zero. Linearization of the diffusion term yields,

$$D_1(\bar{v}^{n+1}) = D_1(\bar{v}^n) + \frac{\partial D_1}{\partial \bar{v}_i} (\Delta \bar{v}_i)^n + \frac{\partial D_1}{\partial \bar{v}_{n(j)}} (\Delta \bar{v}_{n(j)})^n \quad (5.67)$$

where  $\frac{\partial D_1}{\partial \bar{v}_i} = \frac{1 + C_{b2}}{\sigma \text{Re}_L} (\Delta \bar{v} \cdot \underline{n})_{ij} ds_j - \frac{C_{b2}}{\sigma \text{Re}_L} \Delta^2 \bar{v}_i V_i$  (5.68)

$$\frac{\partial D_1}{\partial \bar{v}_{n(j)}} = \frac{1 + C_{b2}}{\sigma \text{Re}_L} (\Delta \bar{v} \cdot \underline{n})_{ij} ds_j \quad (5.69)$$

Linearizing the destruction terms yields

$$D_2(\bar{v}^{n+1}) = D_2(\bar{v}^n) + \frac{\partial D_2}{\partial \bar{v}_i} (\Delta \bar{v}_i)^n \quad (5.70)$$

The Jacobian corresponding to the destruction term is also evaluated numerically as,

$$\frac{\partial D_2}{\partial \bar{v}} = \frac{D_2(\bar{v} + h) - D_2(\bar{v})}{h} \quad (5.71)$$

Substituting Equations (5.62), (5.65), (5.67), and (5.70) into (5.61) and simplifying yield,

$$\begin{aligned} & \left( \frac{V_i}{\Delta t_i} - \frac{\partial C}{\partial \bar{v}_i} - \frac{\partial P}{\partial \bar{v}_i} - \frac{\partial D_1}{\partial \bar{v}_i} - \frac{\partial D_2}{\partial \bar{v}_i} \right) \Delta \bar{v}_i - \left[ \sum_{j=1}^k \frac{\partial D_1}{\partial \bar{v}_{n(j)}} + \sum_{j=1}^k \frac{\partial C}{\partial \bar{v}_{n(j)}} \right] \Delta \bar{v}_{n(j)} = \\ & = C(\bar{v}^n) + P(\bar{v}^n) + D_1(\bar{v}^n) + D_2(\bar{v}^n) \end{aligned} \quad (5.72)$$

The system of equations represented by Equation (5.72) is a sparse matrix system and is solved using the GMRES solver. The GMRES solver is also used for solving the matrix system resulting from equation (5.72).

At the farfield and the inflow boundaries, the value of  $\bar{v}$  is specified such that  $\chi = 5$  (Allmaras [61]) and at the solid boundaries the turbulent viscosity is set to zero.

### 5.13 Geometric Conservation Law

The geometric conservation law adds a correction term to the governing equations to account for the grid motion (Thomas and Lombard [62], Janus [63]). This preserves the uniform conditions when the grid moves, thereby avoiding the creation of spurious sources and sinks in the flow field. The geometric conservation law can be derived from the continuity equation as described below.

Consider a control volume  $\Omega$  with control surface  $\partial\Omega$ , that moves with speed  $\underline{c}$  with respect to a stationary inertial frame. Then the continuity equation can be written in integral form as (Warsi [40]),

$$\frac{d}{dt} \int_{\Omega} \rho d\Omega + \oint_{\partial\Omega} \rho (\underline{V} - \underline{c}) \cdot \underline{n} ds = 0 \quad (5.73)$$

Assuming uniform conditions everywhere in the field, the above equation reduces to

$$\frac{d}{dt} \int_{\Omega} d\Omega = \oint_{\partial\Omega} \underline{c} \cdot \underline{n} ds \quad (5.74)$$

Equation (5.74) represents the geometrical conservation law in integral form. This relates the rate of change of the control volume to the orientation and velocities of the cell faces.

For dynamic grids, a term, accounting for the changes in the control volume, appears in the numerical discretization of governing equations. With this extra term, the semi-discretized form of the governing equations can be written as,

$$V_i^n \frac{\Delta Q_i^n}{\Delta t} + Q_i^n \frac{d}{dt} \int_{\Omega} d\Omega + \int_{\partial\Omega} \underline{F}(Q) \cdot \underline{n} ds = 0 \quad (5.75)$$

The second term in Equation (5.75) is calculated using the geometric conservation law, and is evaluated as

$$Q_i^n \oint_{\partial\Omega} \underline{c} \cdot \underline{n} ds = Q_i^n \sum_{j=1}^k (x_t n_x + y_t n_y + z_t n_z) ds_j \quad (5.76)$$

The above summation is carried over the cell faces. Since the data structure is based on the edges, the above integration is done globally, as in the case of the flux summation.

### 5.14 Grid Movement

For flow simulation over moving bodies, the grid has to be regenerated globally at each time step or the grid points have to be moved appropriately to retain a body conforming grid where grid lines do not cross. Regeneration of grids for each time step is expensive and

interpolation has to be done to transfer the data between the grids. The interpolation is usually non-conservative and it reduces the accuracy of the flow simulation. In the later approach, the grid points are moved using different methods, for example tension spring analogy, potential flow analogy, etc. In the present study, the tension spring analogy is used for grid point movement (Singh et al. [64]).

In tension spring analogy, each edge of the grid is assumed to be a tension spring. The movement of the interior points are calculated by solving the system of tension springs when the boundary points are disturbed. The motion of the boundary points is computed either from the external forces acting on them due to the fluid flow or from a prescribed motion. The spring stiffness is taken as inversely proportional to square of the length of the edge. The spring analogy, by specifying the boundary displacement, will result in a linear system for the interior point displacements. This system is written as

$$\sum_j K_{ij} (\Delta x_i - \Delta x_j) = 0 \quad (5.77)$$

$$\sum_j K_{ij} (\Delta y_i - \Delta y_j) = 0 \quad (5.78)$$

where  $K_{ij}$  is the spring stiffness corresponding to the edge connecting nodes  $i$  and  $j$ , and is defined as  $l_{ij}^{-2}$ , where  $l_{ij}$  is the length of the edge. The summation varies over all nodes that are connected to the node  $i$ . The sparse matrix system resulting from the Equations (5.77) and (5.78) is solved using GMRES, a sparse matrix solver.

For the situations involving large relative motion of the bodies, the quality of the grid will degrade after a few hundred time steps. At this stage a new grid is generated and the solution is transferred to the new grid. A first order interpolation is used to transfer the solution from old grid to the new one. The time marching is restarted with the interpolated solu-

tions on the new grid. Example of the flow simulation over dynamically moving bodies and the capability to handle the grid movement are demonstrated in Section 6.5.

### 5.15 Rigid Body Movement

The trajectory of moving bodies is determined by the laws of classical mechanics. It is assumed that all moving bodies have six degrees of freedom. These six degrees of freedom are the translation along the three coordinate axes and the rotation about the three coordinate axes. The movement of the body is determined based on the pressure distribution over the body due to the fluid flow and the gravitational force due to the weight of the body (Lohner [1], Koomullil et al. [65]).

The notations used for rigid body motion are shown in Figure 5.9. The position vector of any point on the surface of the body is expressed as the vector sum of the position vector of the center of gravity and the vector from center of gravity to the corresponding point (Figure 5.9) and is written as,

$$\underline{r} = \underline{r}_c + \underline{r}_0$$

Now the velocity at the point P is given by,

$$\dot{\underline{r}} = \dot{\underline{r}}_c + \dot{\underline{r}}_0 = \underline{V}_c + \underline{\omega} \times \underline{r}_0$$

where  $\underline{r}$  represents the position vectors shown in Figure 5.9. and the  $\cdot$  represents its derivative with respect to time. The linear velocity and the angular velocity of the body are  $\underline{V}_c$  and  $\underline{\omega}$  respectively.

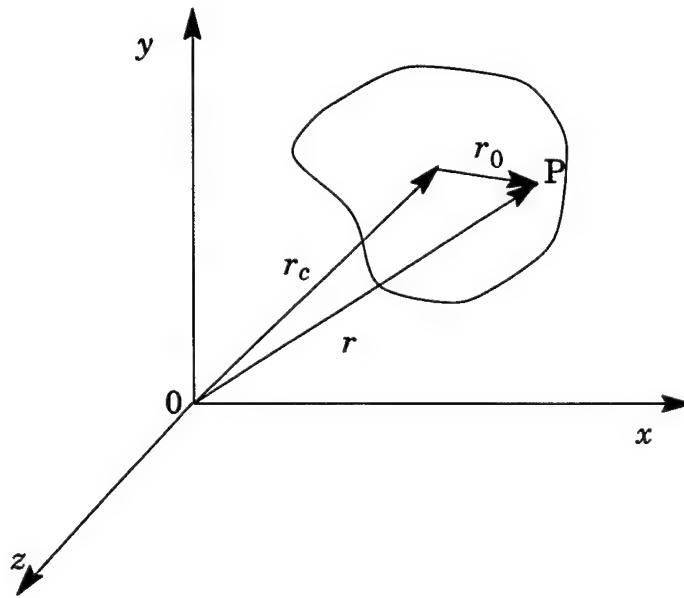


Figure 5.9 Notations Used for Rigid Body Dynamics

The balance of forces and moments acting on the body results in the relations for the translational and rotational accelerations. These expressions are written as

$$m\dot{\underline{V}}_c = \sum \underline{F} = m\underline{g} - \oint_{\partial\Omega} p \underline{n} ds \quad (5.79)$$

$$\Theta \cdot \dot{\underline{\omega}} + \int_{\Omega} (\underline{\omega} \cdot \underline{r}_0) (\underline{r}_0 \times \underline{\omega}) d\Omega = \sum \underline{r}_0 \times \underline{F} = - \oint_{\partial\Omega} p \underline{r}_0 \times \underline{n} ds \quad (5.80)$$

where  $\Theta = \begin{bmatrix} I_{yy} + I_{zz} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{xx} + I_{zz} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{xx} + I_{yy} \end{bmatrix}$

$$I_{ij} = \int_{\Omega} r_o^i r_o^j \rho d\Omega$$

and  $\dot{\underline{\omega}}$  is the angular acceleration vector.

For two dimensional cases, the second term on the left hand side of equation (5.80) vanishes and  $\dot{\underline{\omega}}$  has only one component, which in turn simplifies the equation greatly. For two dimensional cases, equation (5.80) reduces to

$$(I_{xx} + I_{yy}) \dot{\omega}_z = - \oint_{\partial\Omega} p \left( (x - x_c) n_y - (y - y_c) n_x \right) ds \quad (5.81)$$

As the body moves under the action of the aerodynamic and body forces, the second moment of inertia changes. These changes are only due to the rotation of the body, as the body is assumed to be rigid. Figure 5.10 shows the co-ordinate axes before and after the rotation of the body through an angle  $\theta$  with respect to the center of gravity. The moments of inertia with respect to the new co-ordinate axes are related to that with respect to the old co-ordinate axes by (Fletcher [66]),

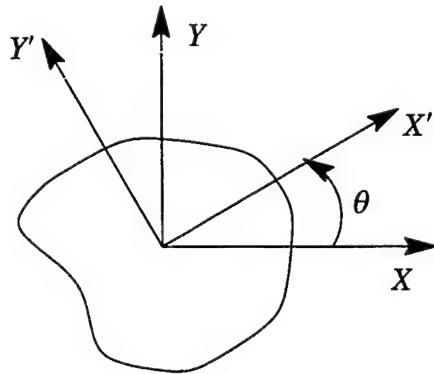


Figure 5.10 Orientation of Reference Axis and Rotated Axis

$$I_{x'x'} = I_{xx} \cos^2 \theta + I_{yy} \sin^2 \theta + 2 I_{xy} \sin \theta \cos \theta \quad (5.82)$$

$$I_{y'y'} = I_{xx} \sin^2 \theta + I_{yy} \cos^2 \theta - 2 I_{xy} \sin \theta \cos \theta \quad (5.83)$$

$$I_{x'y'} = (I_{xx} - I_{yy}) \sin \theta \cos \theta + I_{xy} (\cos^2 \theta - \sin^2 \theta) \quad (5.84)$$

The moments of inertia are updated after each time step.

## 6. Results and Discussions

### 6.1 Validation of Two Dimensional Flows

Two dimensional Euler calculations are validated by computing the flow over the NACA0012 airfoil. The results are compared with the experimental data. The grid used for this simulation consists of 1961 nodes and 3332 cells (Figure 6.11). The freestream Mach number is 0.63 and the angle of attack is 2.0 degrees.

The results from the first and second order accurate, inviscid calculations are compared with the experimental data (AGARD) in Figure 6.12. This plot shows a noticeable improvement in accuracy for the second order calculations compared to the first order calculations. The results from the second order computations agree well with the benchmark data.

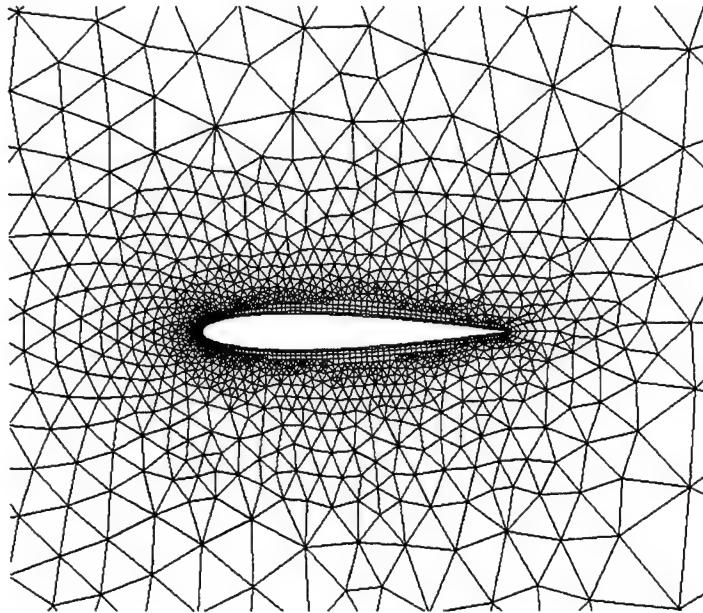


Figure 6.11 Hybrid Grid Around NACA0012 Airfoil

The convergence history associated with the implicit and explicit calculations for the testcase described above is given in Figure 6.13. The implicit scheme converges to the steady state within 300 time steps, while the explicit scheme takes 25000 iterations to re-

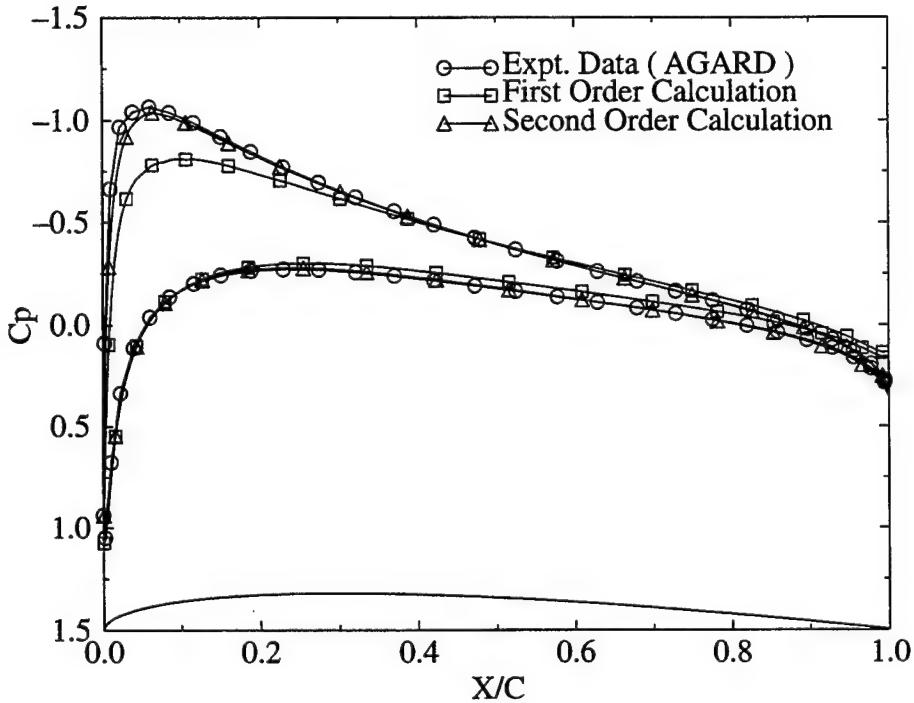


Figure 6.12 Cp Distribution Over NACA0012 Airfoil at Freestream Mach Number 0.63 and Angle of Attack 2.0 Degrees

duce the L2 norm of the residual to ten orders of magnitude. The L2 residual is defined as

$$\| L2 \| = \log_{10} \left[ \sqrt{\sum_{i=1}^{ncells} \sum_{j=1}^4 (\Delta Q_{ij})^n} \right]$$

Numerical Jacobians are used in the implicit calculations with a CFL of 50 and the explicit calculations are performed with a CFL of 1.5. The explicit scheme takes 12813.76 seconds for 25000 iterations on a single processor of an SGI R8000, while the implicit scheme takes 1443 seconds for 300 time steps. Thus the time requirement for the simulation can be reduced by an order of magnitude by using an implicit scheme.

The laminar flow calculations are validated using the flow over a flat plate at a freestream Mach number of 0.5 and a Reynolds number of 30000. The grid consists of

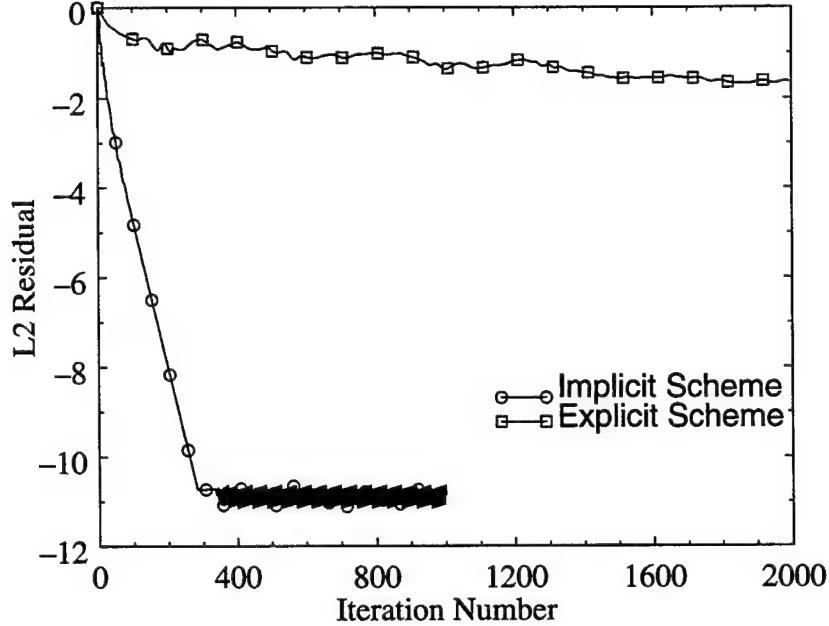


Figure 6.13 Convergence History for Explicit and Implicit Schemes for NACA0012 Airfoil at Freestream Mach Number 0.63 and Angle of Attack 2.0 Degrees

2911 nodes and 2800 rectangles. The physical domain includes five plate lengths upstream of the leading edge, four plate lengths downstream of the trailing edge, and five units above the plate. The first point off the plate is at a distance of  $1.0 \times 10^{-4}$ . The computed velocity profile is compared with the Blassius velocity profile (Warsi [40]), and is presented in Figure 6.14. The non-dimensional  $\eta$  axis is defined as,

$$\eta = y \sqrt{\frac{Re_L}{2 \nu x}}$$

The computed velocity profiles are plotted in Figure 6.14 for axial location 0.4, 0.5, and 0.6 and these profiles are independent of the axial location. Therefore, they satisfy the principle of self similar boundary layers (Warsi [40]).

The turbulent flow simulation is validated using the flow over a flatplate. The dimensions of the flow domain are identical to that used for the laminar flow simulation, while using a finer distribution of points near the flat plate. The first point of the plate is at a distance of  $1.0 \times 10^{-5}$ . The freestream Mach number for this simulation is taken as 0.5

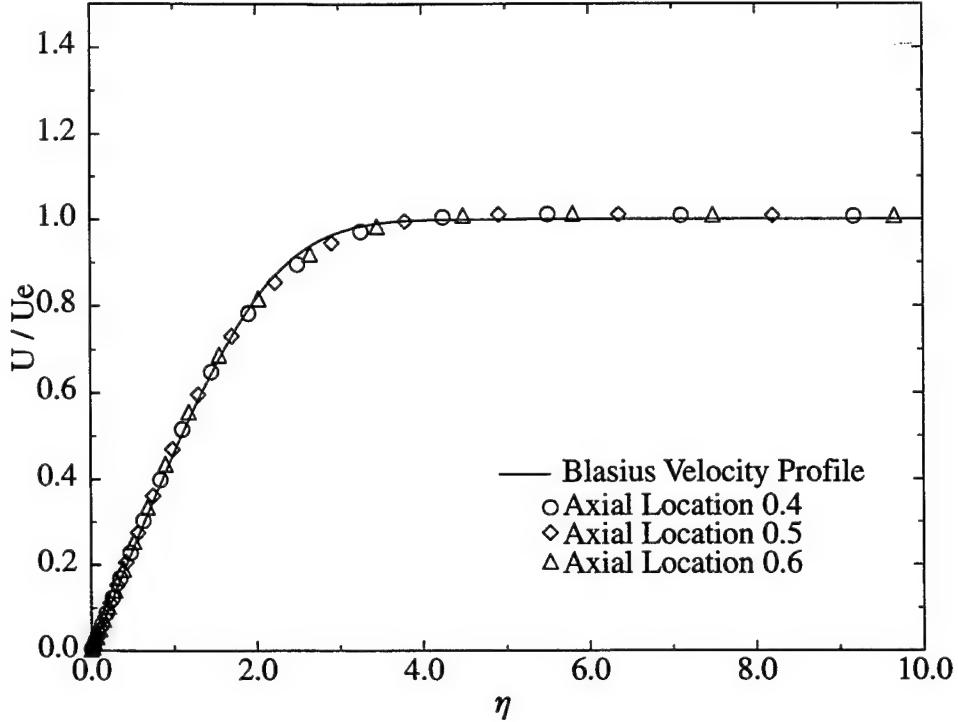


Figure 6.14 Laminar Velocity Profile Comparison With Theoretical Data

and a Reynolds number of  $2 \times 10^6$  was used. The computed turbulent velocity profile is compared to theoretical results in Figure 6.15.

The next geometry used for the implementation of the turbulence model is the standard NACA0012 airfoil. The flow conditions used were a freestream Mach number of 0.799, an angle of attack of 2.26 degrees and a Reynolds number  $9.0 \times 10^6$ . The grid used for this simulation is a structured grid of dimension 290x81 (Figure 6.16) and is converted into the hybrid grid data structure before the simulation. A total of 200 points were given on the surface of the airfoil. The Mach number contours are shown in Figure 6.17.

The computed pressure coefficient is compared with the experimental data (Baldwin and Barth [67]) and the results from Baldwin-Barth computations (Baldwin and Barth [67]). This is demonstrated in Figure 6.18. Slight deviation of the computed results as compared to the experimental values is realized in the simulation.

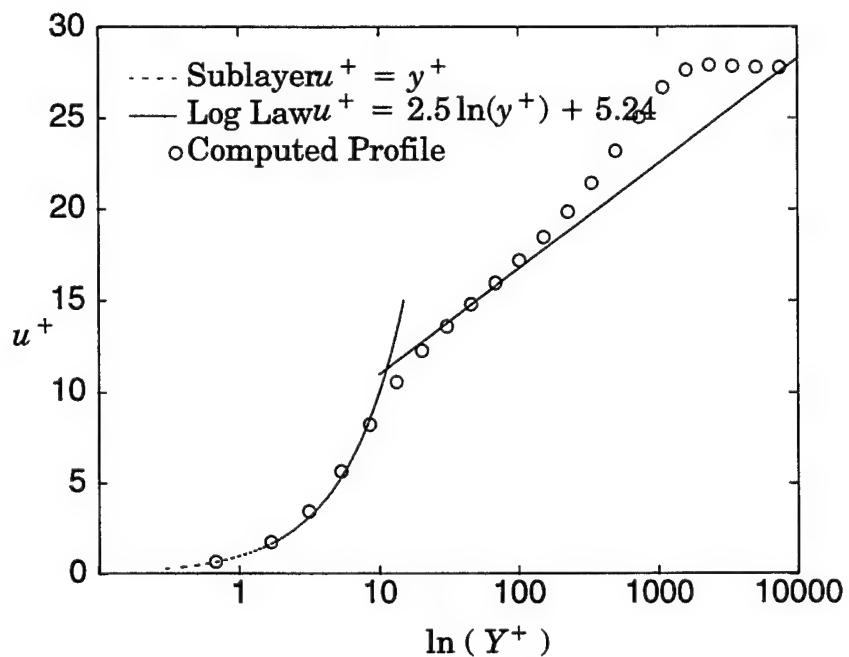


Figure 6.15 Turbulent Velocity Profile for Reynolds Number 2.0E+06 Compared with Theoretical Data

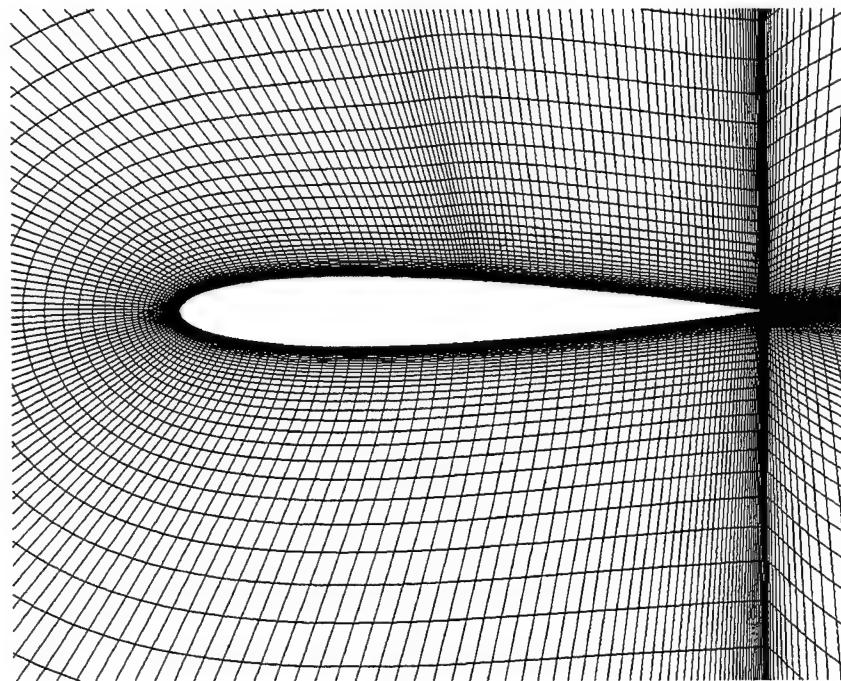
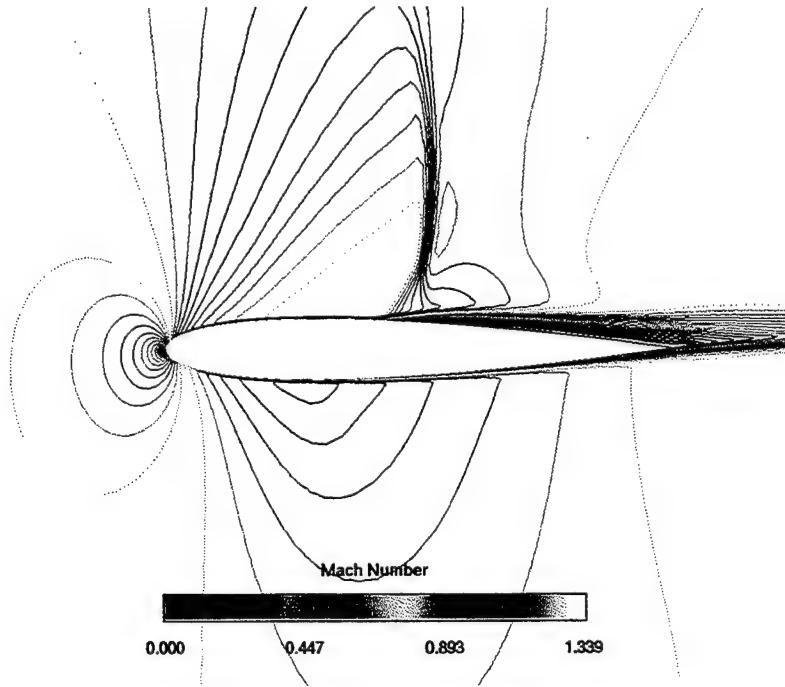


Figure 6.16 Grid Used for Turbulent Flow Over NACA0012 Airfoil

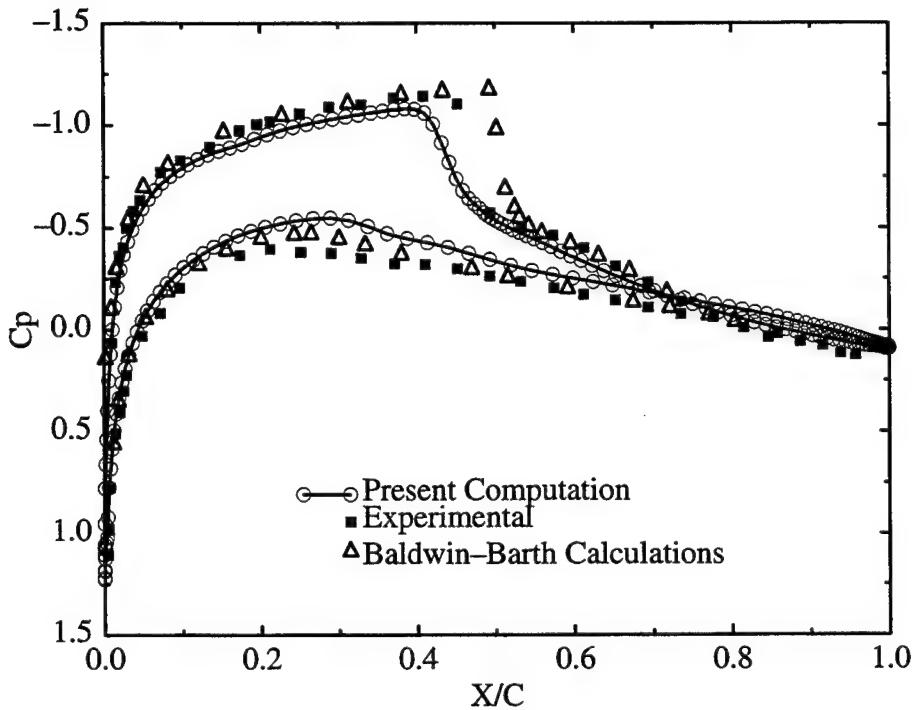


**Figure 6.17 Mach Number Contours for NACA0012 Airfoil at Freestream Mach Number 0.799 and Angle of Attack 2.26 Degrees**

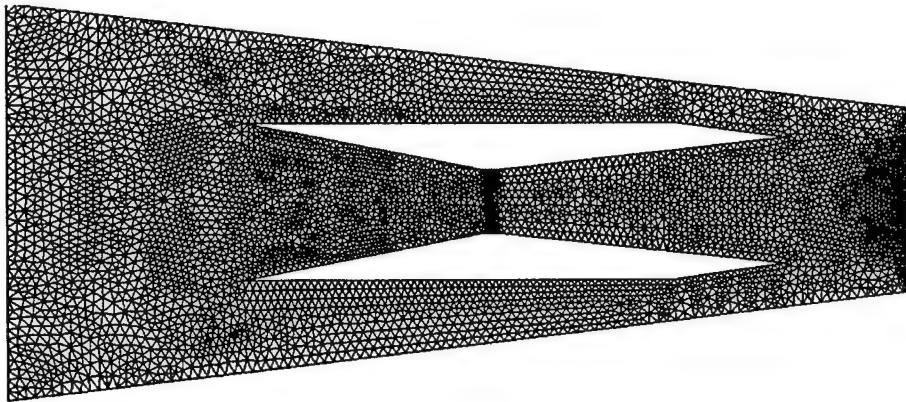
The memory requirement for double precision calculations is approximately 595 words per cell for implicit scheme and 102 words per cells for explicit schemes. Not much effort has been put towards the optimization of the memory requirements.

## 6.2 Effect of Limiter Function on Convergence

The effect of different limiter functions on the convergence of solutions to steady state is discussed in the following section. The geometry considered for this study is a scramjet engine (Grimes [68]). The grid used for these calculations is shown in Figure 6.19, and consists of 5515 points and 10404 triangles. The freestream Mach number and the angle of attack for this testcase were 3.5 and 0 degree respectively. The pressure distribution inside the scramjet engine, obtained from the simulation, is shown in Figure 6.20.



**Figure 6.18 Pressure Coefficient Comparison over NACA0012 Airfoil at Freestream Mach Number 0.799 and Angle of Attack 2.26 Degrees**



**Figure 6.19 Unstructured Grid for Scramjet Like Geometry.**

Convergence history for this simulation with different limiter function options is plotted in Figure 6.21. The residual stalls after two orders of magnitude drop in the case of Barth limiter. However, residual drops to the machine zero with a higher threshold coefficient  $K$  for Venkatakrishnan's limiter. The threshold factor  $K$  has a strong influence on

convergence, as can be seen in Figure 6.21. For larger values of the coefficient  $K$ , the limiter function is limited to places with high gradients. By setting the threshold coefficient  $K$  to zero, the limiter is active everywhere. With a proper choice of the threshold factor, Venkatakrishnan's limiter gives a better convergence than Barth's limiter.

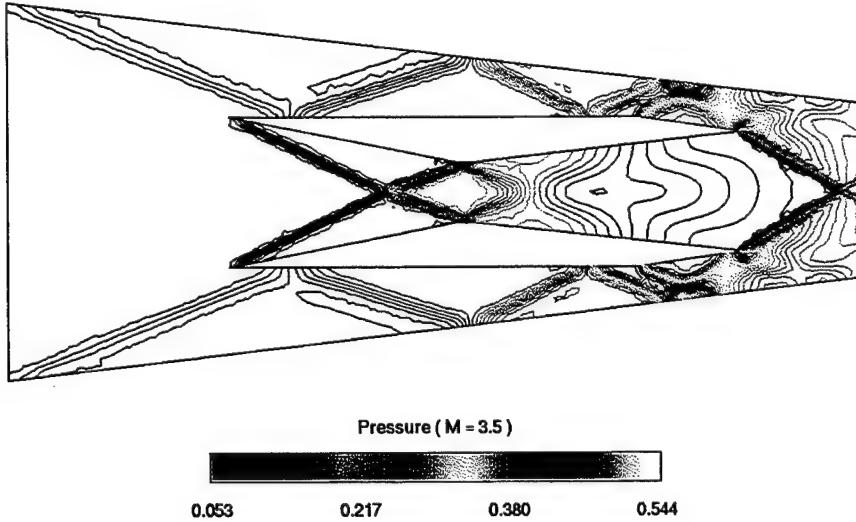


Figure 6.20 Pressure Distribution in Scramjet Engine for Freestream Mach Number 3.5

The computed skin friction distribution over the flat plat for a Reynolds number of  $2 \times 10^6$  and a freestream Mach number of 0.5 is compared with the theoretical values in Figure 6.22. The theoretical values are calculated using the following expression (Warsi [40]).

The Venkatakrishnan's limiter under predicts the the skin friction, but the Barth limiter gives a non-smooth skin friction coefficients. For very high values of the threshold parameter ( $K=50$ ) the distribution of the skin friction deviates from theoretical ones.

The effect of the limiter function on the skin friction coefficient is studied using the flow over the NACA0012 airfoil, which is used for the validation of the turbulence flow calculation. The skin friction distribution over the airfoil is plotted in Figure 6.23.

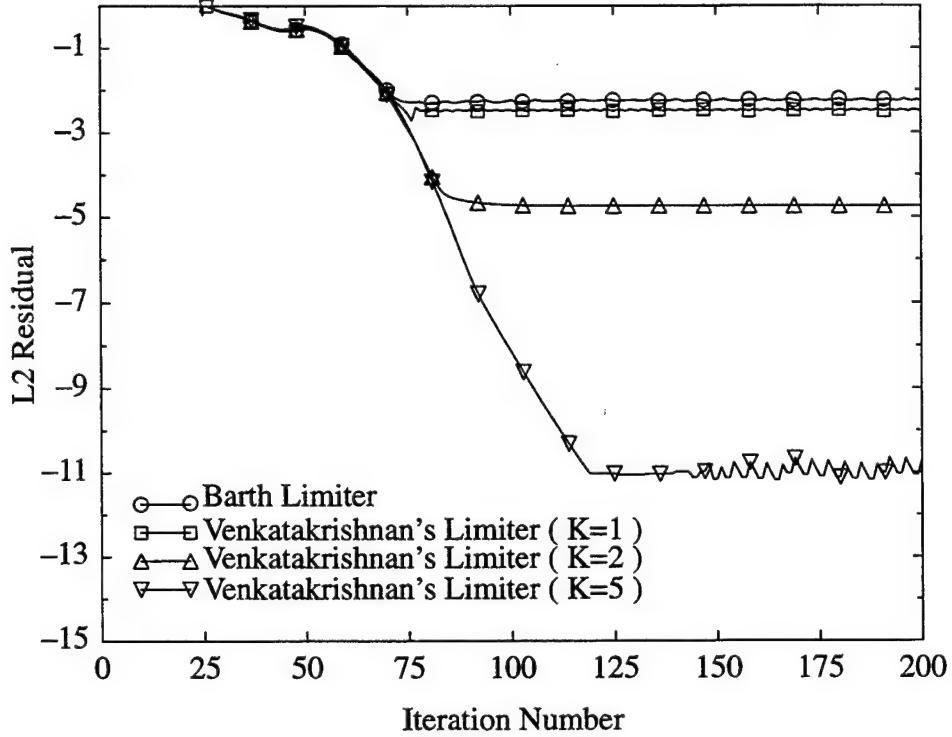


Figure 6.21 Effect of Limiter on Convergence to Steady State Solution

$$c_f = 0.02628 \text{ } Re_x^{-1/7}$$

The skin friction distribution does not change with different threshold parameters for Venkatakrishnan's limiter. But Barth limiter gives higher skin friction at the leading edge of the airfoil and lower skin friction after the leading edge. After the shock induced separation at the upper surface of the airfoil both the limiters gives the same skin friction except for K=20.

### 6.3 Effect of Approximate and Numerical Jacobians On Convergence

The scramjet engine geometry, shown in Figure 6.19, is used to study the effect of approximate analytical and numerical Jacobians on the convergence of solution to steady state. The freestream Mach number of 3.5 and an angle of attack of 0.0 degree were used for this simulation. The convergence history for this testcase with different Jacobians is plotted in Figure 6.24 and Figure 6.25. Figure 6.24 shows the convergence history for a CFL number of 5. It can be seen that there is not much advantage in using the numerical

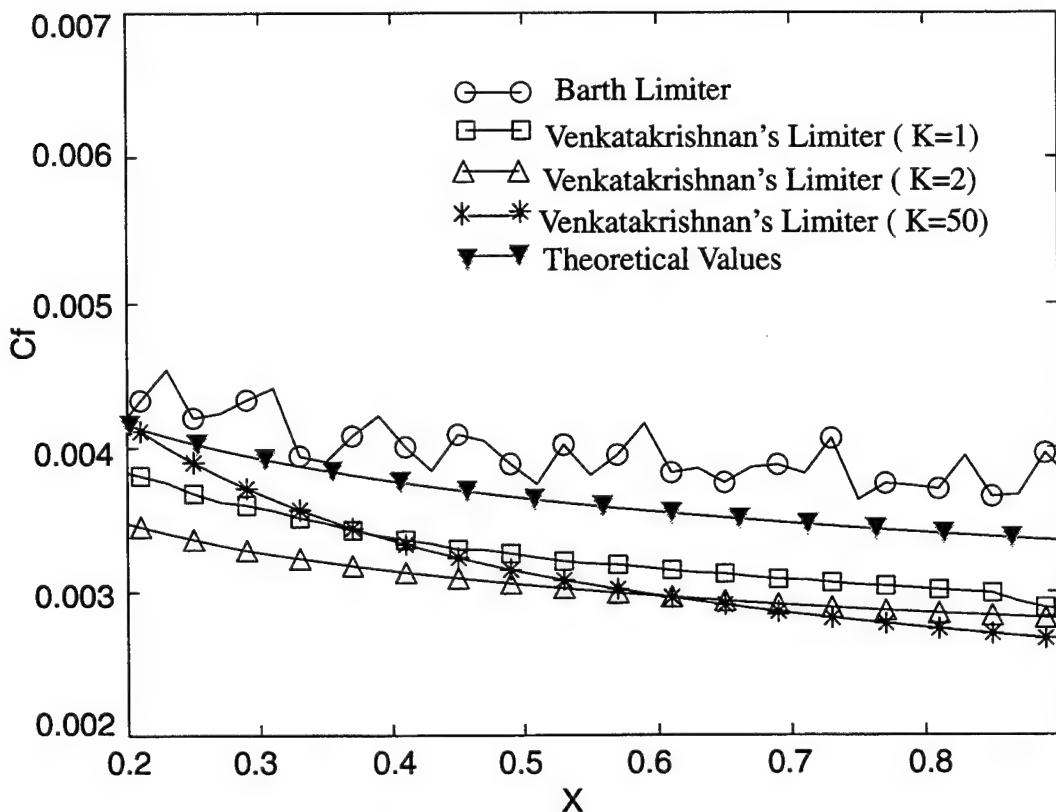
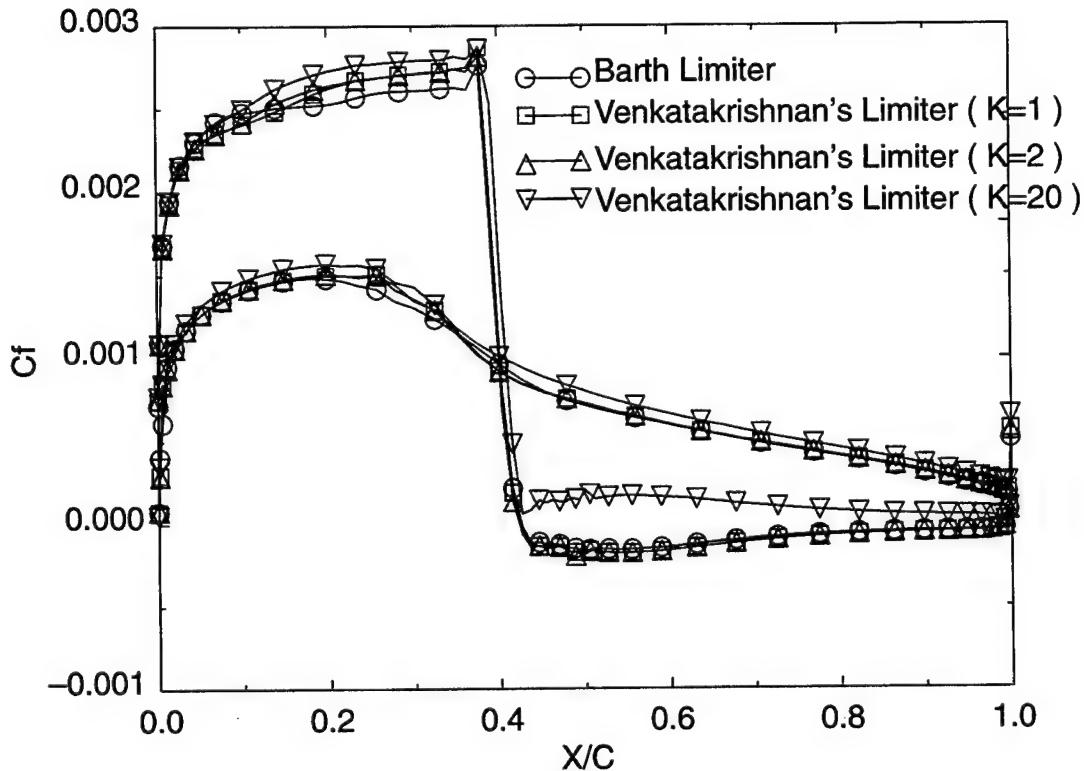


Figure 6.22 Skin Friction Distribution Over a Flat Plate at Reynolds Number  $2 \times 10^6$  and Freestream Mach Number 0.5

Jacobian compared to the approximate Jacobian. When the CFL number is increased to 20, the solution converges much faster for the numerical Jacobian compared to the approximate analytical Jacobian as shown in Figure 6.25. The numerical Jacobian calculations take approximately 5% more CPU time per iteration than the approximate analytical Jacobian. But the total time for the simulation is less for the numerical Jacobians because of the faster convergence to the steady state. The memory requirements are the same for both the approaches. During the time integration the Jacobians are evaluated at each time level.

The same behavior is noticed for the subsonic flow over the NACA0012 airfoil. The freestream Mach number and angle of attack for this simulation are 0.63 and 2.0 degrees respectively. The convergence history for the implicit scheme with approximate analytical and numerical Jacobians are plotted in Figure 6.26. These computations were



**Figure 6.23 Skin Friction Coefficient Distribution Over NACA0012 Airfoil at Freestream Mach Number 0.799 and Angle of Attack 2.26 Degrees**

performed at a CFL of 50. From Figure 6.26, it is evident that the numerical Jacobian gives a better convergence rate than the approximate analytical Jacobian.

For the Navier–Stokes calculations, the numerical Jacobian does not have a significant advantage over the approximate analytical Jacobian. But the solution process employing numerical Jacobians is more stable compared to that using the approximate analytical Jacobians. This is demonstrated using the convergence history for the laminar flow over the flat plate in Figure 6.27. The implicit scheme with approximate analytical Jacobian was not stable at a CFL of 20.

#### 6.4 Examples of Mixed Element Type Grids

Laminar flow simulations were carried out for a four element airfoil at a free-stream Mach number of 0.201 and an angle of attack of 0.0 degree. The Reynolds number

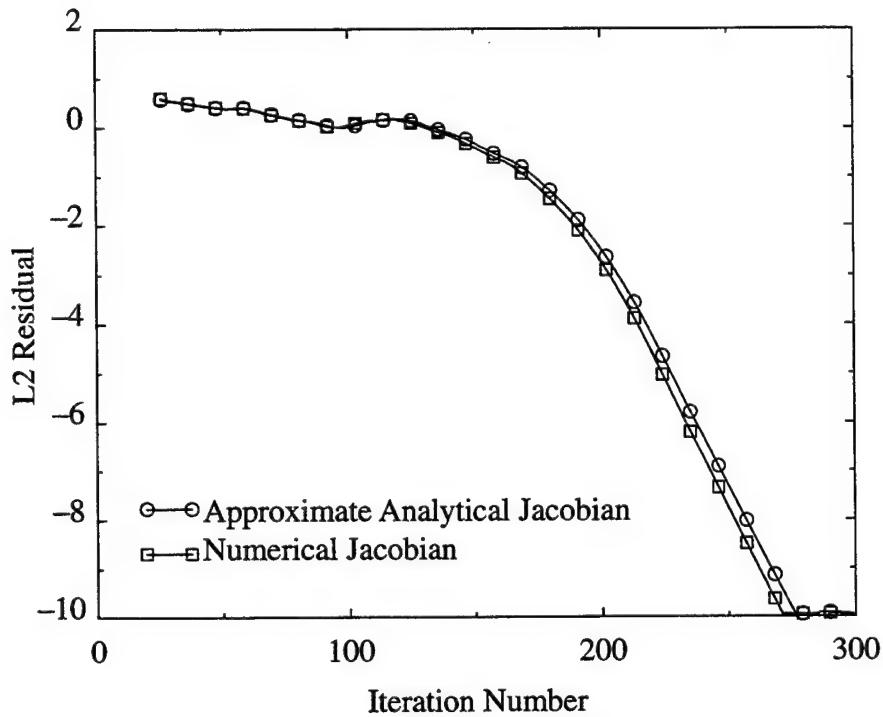


Figure 6.24 Convergence History for Scramjet Geometry Using Approximate Analytical and Numerical Jacobians for CFL 5

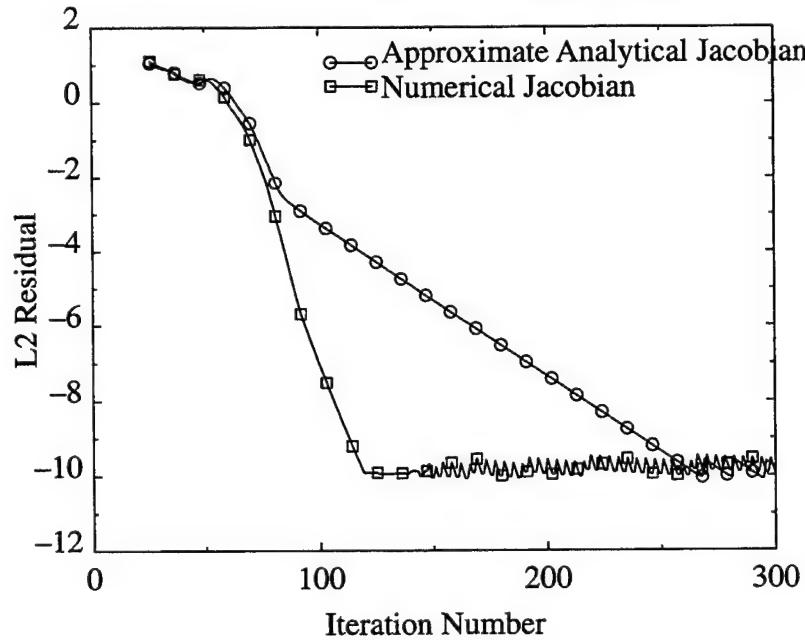
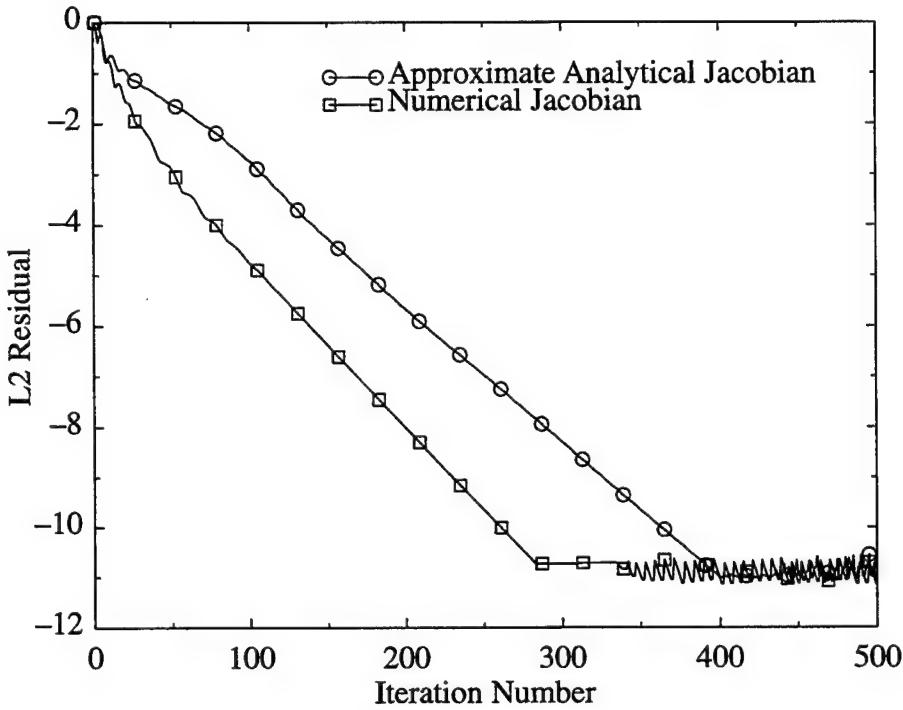


Figure 6.25 Convergence History for Scramjet Geometry Using Approximate Analytical and Numerical Jacobians for CFL 20



**Figure 6.26** Convergence History for NACA0012 Airfoil at Freestream Mach Number 0.63 and Angle of Attack 2.0 Degrees for CFL 50

based on the chord length of the main body of the airfoil was taken as  $2.0 \times 10^5$ . The grid used for this simulation is given in Figure 6.28. It consists of 10445 nodes and 15603 cells, of which 5274 cells have four sides. The pressure and Mach number distributions for this case are given in Figure 6.29.

The capability of handling the cells with an arbitrary number of sides in the flow field is demonstrated using the grid shown in Figure 6.30. This case involves cells with four, five, and six sides. It contains 2345 nodes and 2536 cells. Some of the cells have hanging nodes, where only two edges are connected to that node. The flow conditions taken were a freestream Mach number of 0.30 and an angle of attack 0.0 degree. The pressure contours for this simulation along with the grid is shown in Figure 6.31.

### 6.5 Unsteady Flow Simulation

The unsteady calculations are validated using inviscid transonic flow over NACA0012 airfoil pitching about the quarter chord. A structured grid, shown in

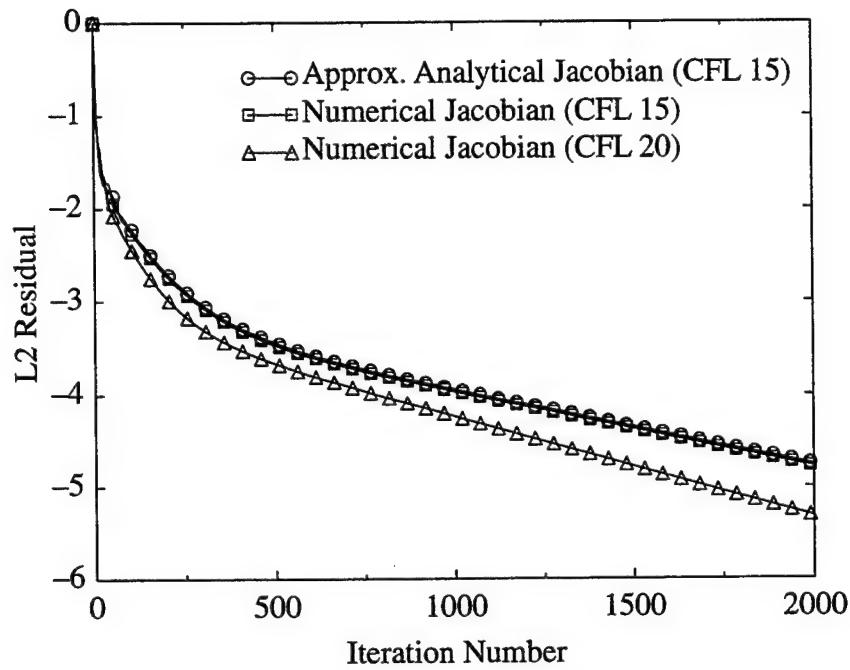


Figure 6.27 Convergence History for Flat Plate Using Approximate Analytical and Numerical Jacobians at Freestream Mach Number 0.5 and Reynolds Number 30000

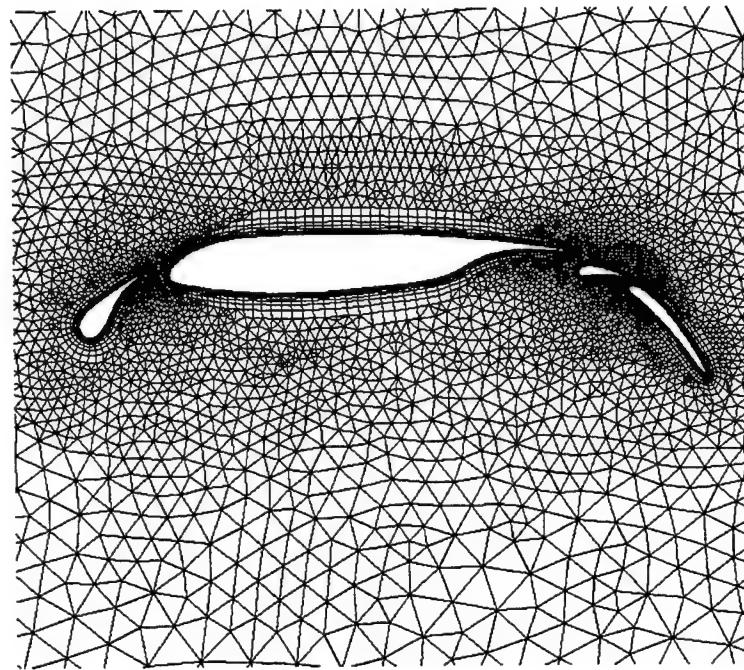
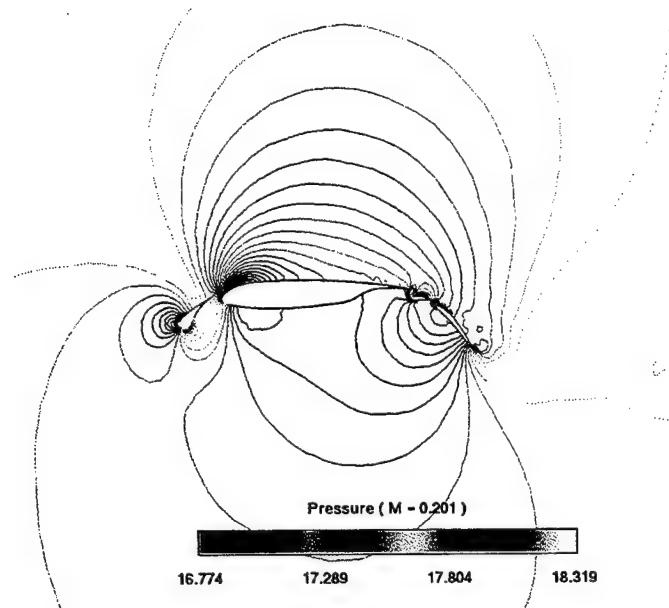
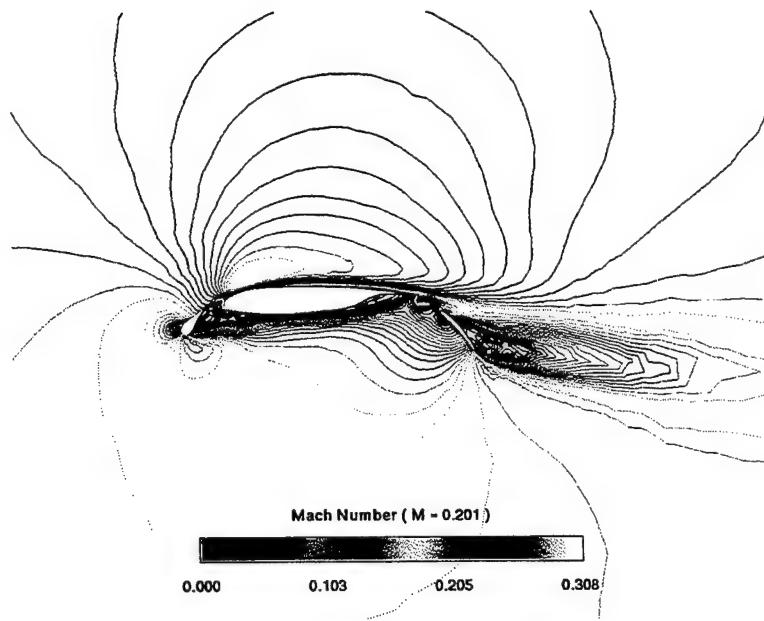


Figure 6.28 Hybrid Grid for Four Element Airfoil



(a) Pressure Distribution



(b) Mach Number Distribution

Figure 6.29 Pressure and Mach Number Distributions for a Four Element Airfoil at Freestream Mach Number 0.201 and Angle of Attack 0 degree

Figure 6.32, with 6000 cells and 6211 nodes is used for this simulation. It is converted into the hybrid grid data structure format so that it can be used with the solver developed in this study. The unsteady calculations were started from a converged steady state solution.

The movement of the airfoil is prescribed such that the angle of attack varies sinusoidally according to the following relation

$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t)$$

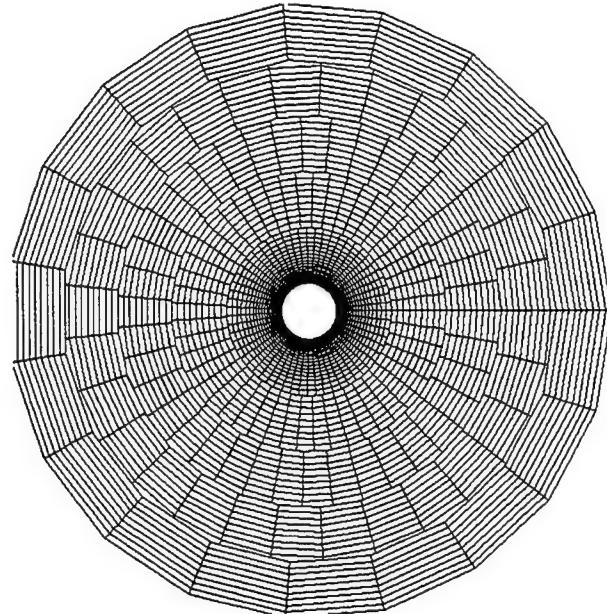
where  $\alpha_m$  is the mean angle of attack and is 0.016 degrees and the maximum deflection  $\alpha_0 = 2.51$  degrees. The NACA0012 airfoil is assumed to be pitching at a reduced frequency,  $k$ , of 0.1628 at a freestream Mach number of 0.755. The reduced frequency is defined as,

$$k = \frac{\omega c}{V_\infty}$$

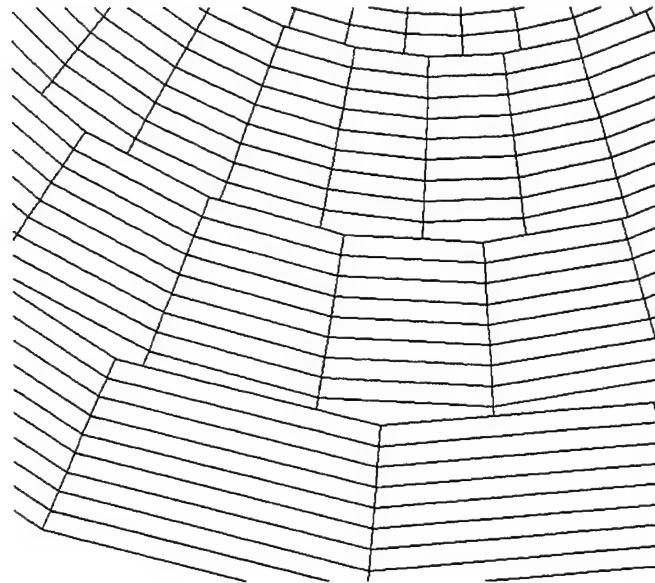
where  $\omega$  is the frequency in radians per second,  $c$  is the chord length, and  $V_\infty$  is the freestream velocity. The computed lift history is plotted in Figure 6.33. The results are compared with the experimental data of Landon[69]. The lift history attains the periodic nature after half a cycle of oscillation.

The pressure over the airfoil at two different angles of attack are presented in Figure 6.34. Figure 6.34(a) gives the pressure distribution at an angle of attack 0.48 degrees when it is moving up. Figure 6.34(b) plots the pressure distribution at an angle of attack 0.98 degrees when the airfoil is pitching down. Even though the angles of attack are almost the same, the shock location differs due to the unsteadiness of the flow.

The residual history is plotted in Figure 6.35. The residual achieves a periodic state after 50 iterations. This simulation was done using a CFL of 1000 and is kept constant during the simulation and it took 50 minutes of CPU time on a single R4400 SGI processor for one period of oscillation. The coefficients of lift and drag generated during the pitching of the airfoil along with the angle of incidence are plotted in Figure 6.36.



(a) Overall View of the Grid



(b) Zoomed View at the Interface Between Layers

**Figure 6.30 Generalized Grid Containing More Than Four Nodes per Cell**

The above described pitching of the airfoil was also simulated on a hybrid grid shown in Figure 6.11. The computed lift history versus the angle of incidence is plotted in

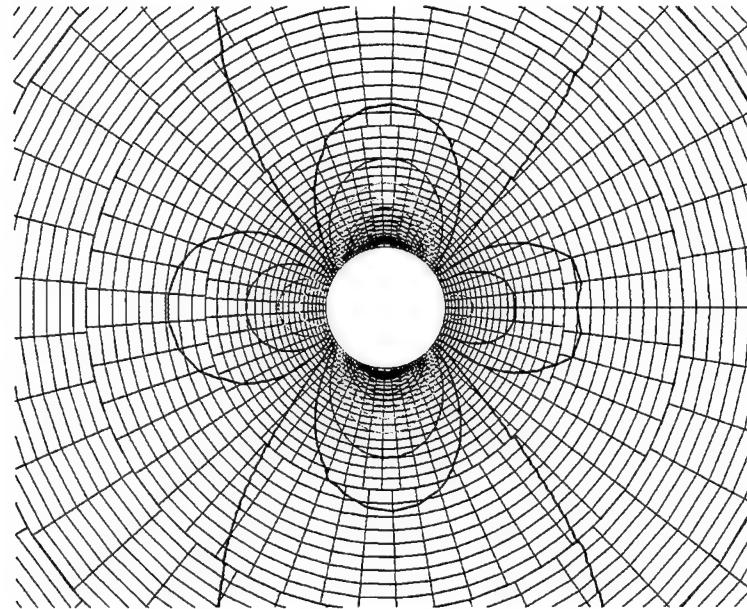
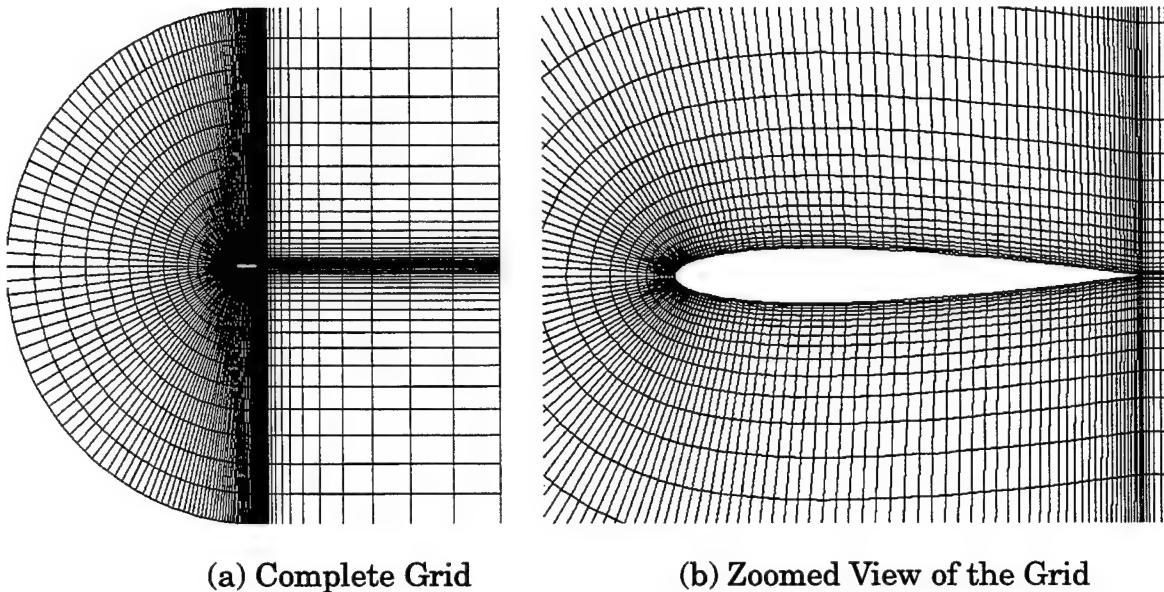


Figure 6.31 The Contour Plot of Pressure Distribution Around Circular Cylinder at Freestream Mach Number 0.3



(a) Complete Grid

(b) Zoomed View of the Grid

Figure 6.32 Grid Used for Pitching Airfoil

Figure 6.37. There is a slight difference between the computed solutions using the hybrid and structured grids. This is attributed to the difference in the grid resolution between the two grids. Figure 6.33–Figure 6.36 are plotted using the simulation done on the grid shown in Figure 6.32.

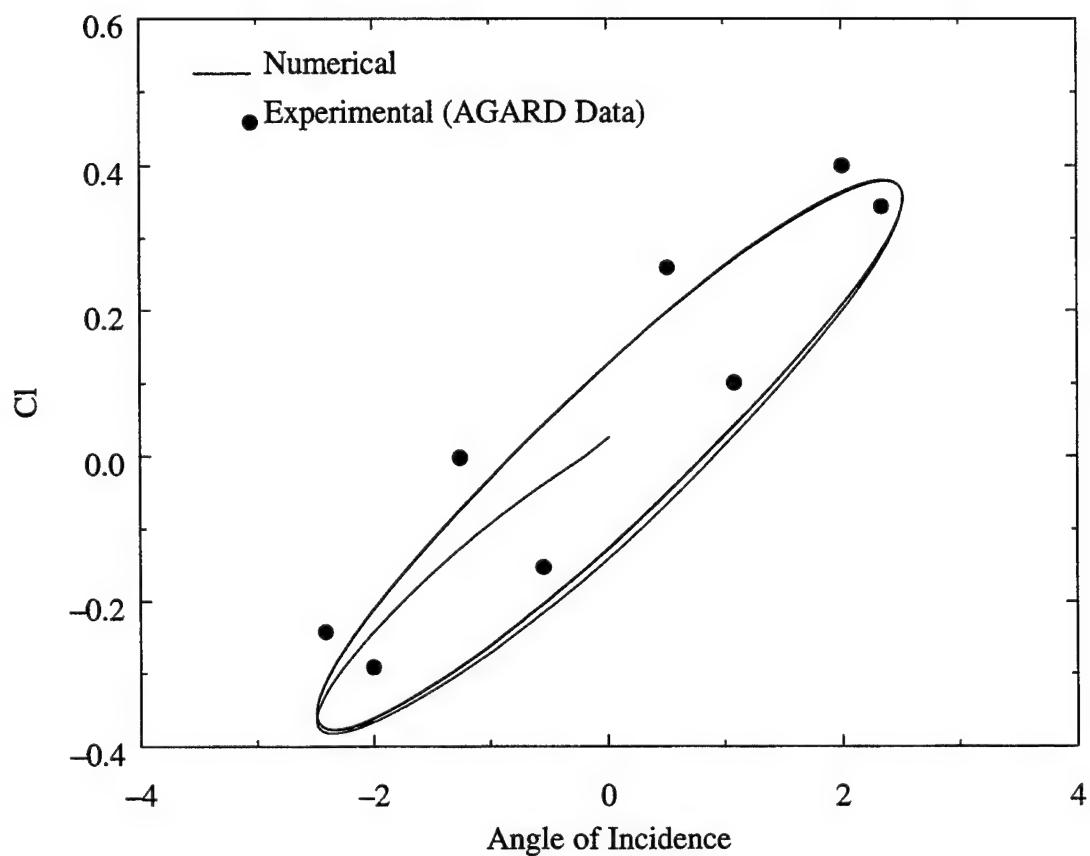


Figure 6.33 Lift History for Pitching NACA0012 Airfoil

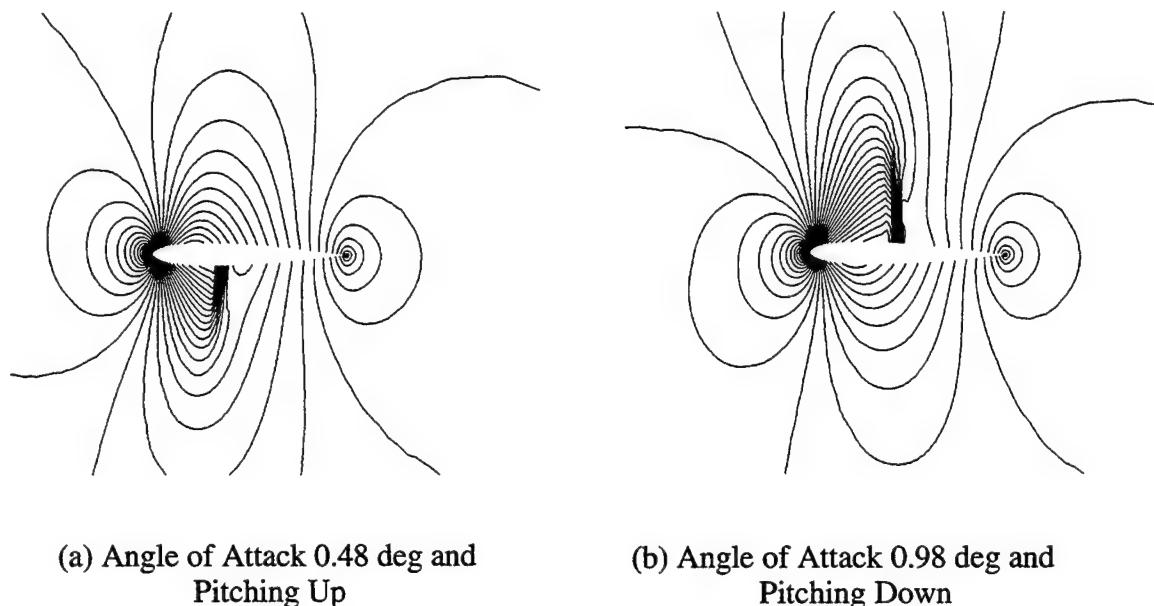


Figure 6.34 Pressure Distribution Around A Pitching NACA0012 Airfoil

### 6.6 Example For Bodies in Relative Motion

Capability to do flow simulations involving bodies in relative motion is demonstrated using a missile separation from a solid body. The geometry consists of a flat solid surface and a missile. The grid used for this simulation consists of 4824 nodes and 9216 cells. The freestream Mach number is taken as 2.0. The unsteady simulation is started from a steady state solution. The pressure and Mach number distributions for the steady state condition are shown in Figure 6.38. The trajectory of the missile is calculated by using the aerodynamic forces acting on it and its weight. During each time iteration, the position of the body changes and the corresponding changes in the interior grid points are calculated using the spring analogy described in Chapter V. The grids are regenerated when the cells get too skewed and the solution is interpolated to the new grid. Regeneration was carried out three different times during the solution process. The grid, pressure distribution, and Mach number distribution at two different time levels are shown in Figure 6.39 and Figure 6.40.

### 6.7 Validation For Three Dimension Flow Simulation

Flow simulations are performed over an ONERA M6 wing at a freestream Mach number of 0.84 and an angle of attack of 3.06 degrees. The Euler calculations are carried out on a structured grid of size 97x25x17. Two different views of the grid are shown in Figure 6.41. A structured grid is used for this simulation to compare the results with those from an existing code (CFL3D (Thomas et al. [22])). The flow simulations are accomplished on the same grid using both the CFL3D code and the present program under validation. Then the results of computations are compared with the experimental data in Figure 6.42. It shows the  $C_p$  distribution over the upper and lower surfaces of the wing at 45%, 65% and 90% span. The calculated values agree well with the experimental results and CFL3D simulation.

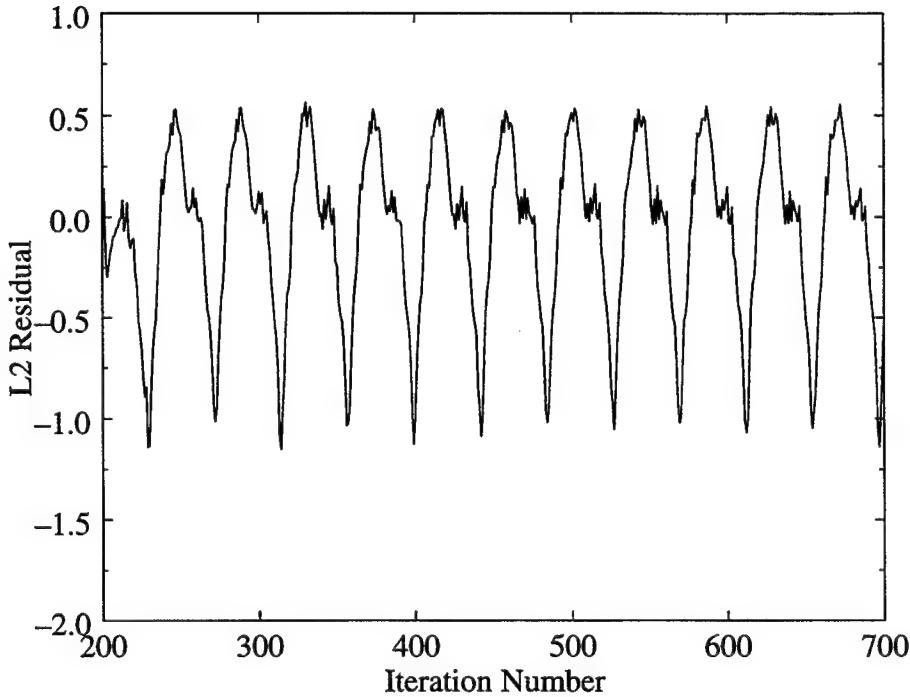


Figure 6.35 Residual History for Pitching NACA0012 Airfoil

Flow simulation was performed for a complete F-15 fighter aircraft with faired inlet and nozzle. The freestream conditions were a Mach number of 0.9 and an angle of attack of 4.84 degrees. The unstructured grid was generated at McDonnell Douglas Aerospace, Saint Louis. A coarser distribution of grid points is used for the fuselage and a fine distribution over the wing surface (Figure 6.43). The grid consists of 44,272 nodes and 235,363 tetrahedra with 6404 nodes and 12,669 faces on the surface.

A four stage Runge–Kutta scheme is used for this simulation because of the unavailability of memory in the local machine for the implicit scheme. Four orders of reduction in the L2 norm of the residual is taken as the criteria for attaining the steady state solution. The convergence history is plotted in Figure 6.44. The pressure distribution on the body surface together with the grid on the symmetry plane is shown in Figure 6.45 and the Mach number contours on the entire aircraft surface are in Figure 6.46.

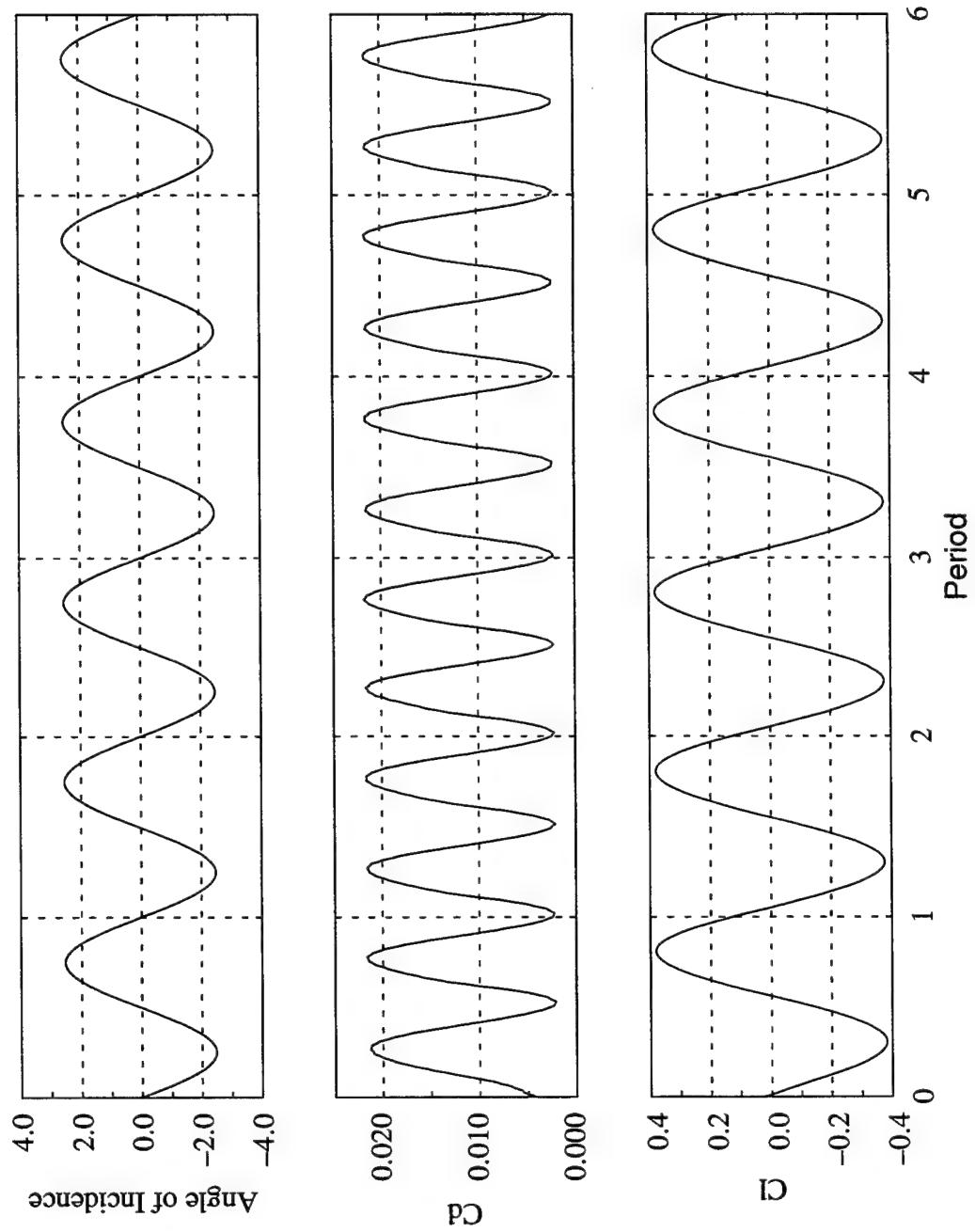


Figure 6.36 Coefficients of Lift and Drag Generated During the Pitching Motion of the Airfoil

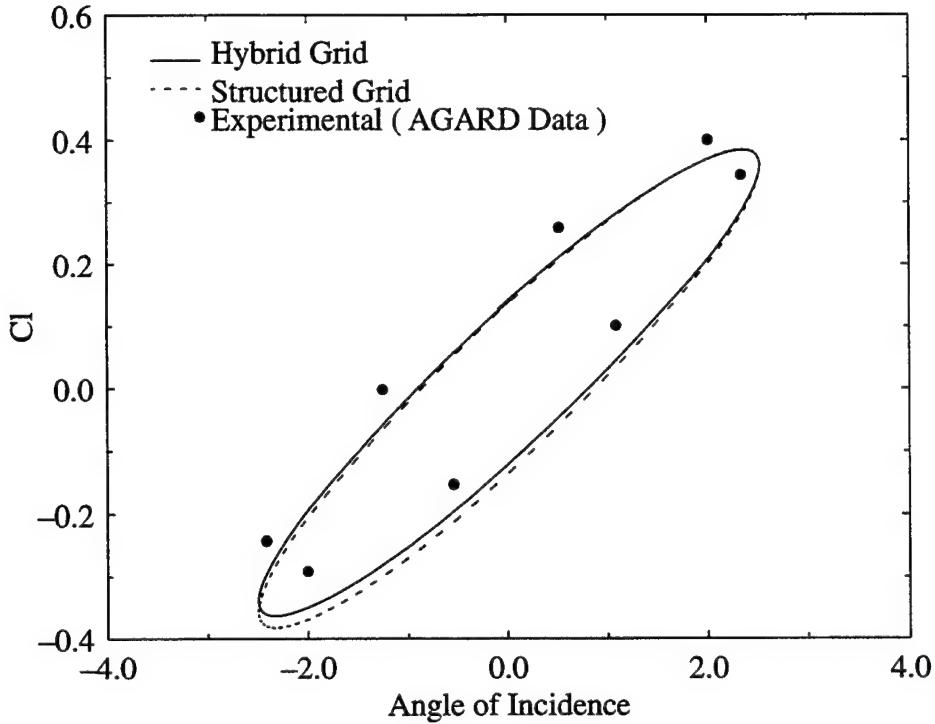


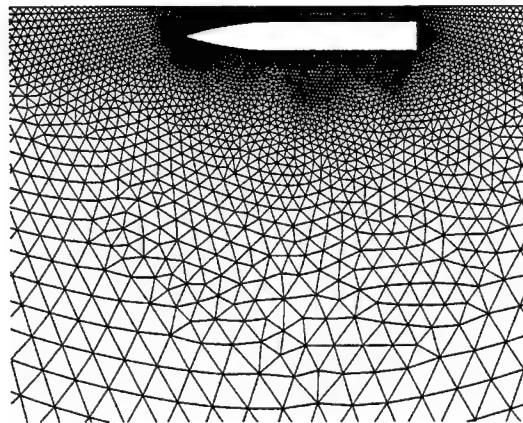
Figure 6.37 Lift History for Hybrid and Structured Grids

The calculated pressure distribution on the surface is compared with the available experimental data ( Michal and Halt [70] ). The pressure distributions at 36% and 59% span of the wing are plotted together with the experimental data in Figure 6.47.

### 6.8 Preliminary Work On Parallelization

The main drawback of the flow solvers using unstructured algorithms is the high memory and CPU requirements. Parallelization of the flow solver provides a means of overcoming this disadvantage. Preliminary work has been done towards the parallelization of the hybrid flow solver developed during this study. Initial results show the potential benefits of parallelization of the flow solver.

The partitioning of the grid for parallel processing is done using a public domain software, METIS ( Karypis and Kumar [71] ), for graph partitioning. The graph corresponding to the grid is given as an input to the software and is divided into a number of



(a) Grid Used

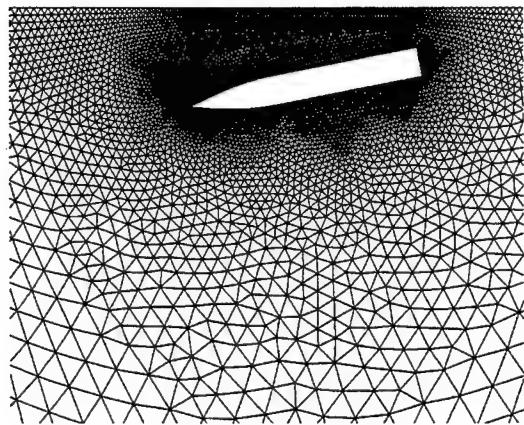


(b) Pressure Distribution

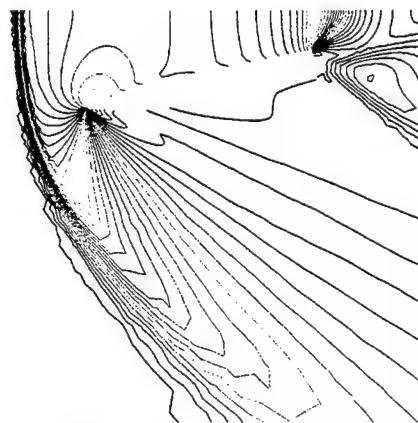


(c) Mach Number Distribution

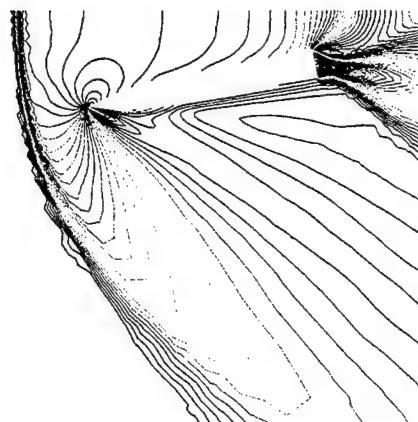
Figure 6.38 Steady State Solution For Missile Geometry  
at Freestream Mach Number 2



(a) Grid

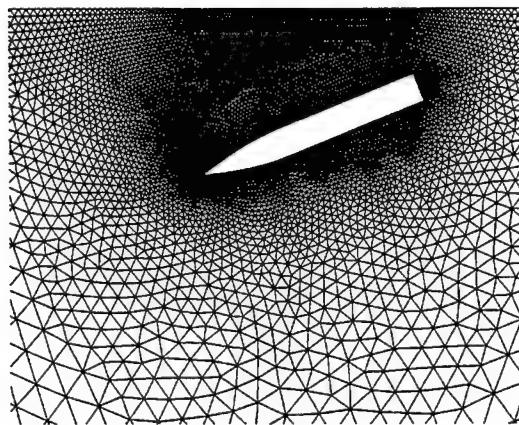


(b) Pressure Distribution

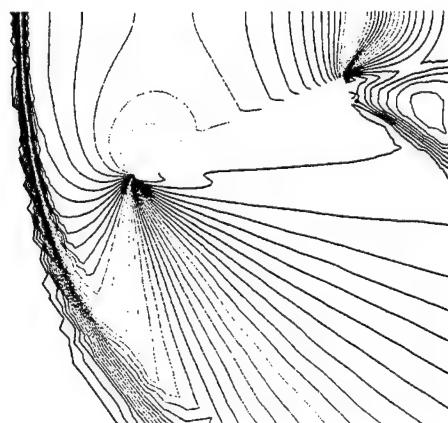


(c) Mach Number Distribution

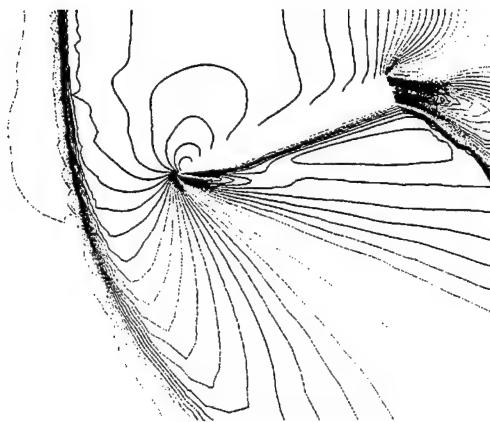
Figure 6.39 Grid, Pressure Distribution and Mach Number Distribution at  $t=177.52$



(a) Grid

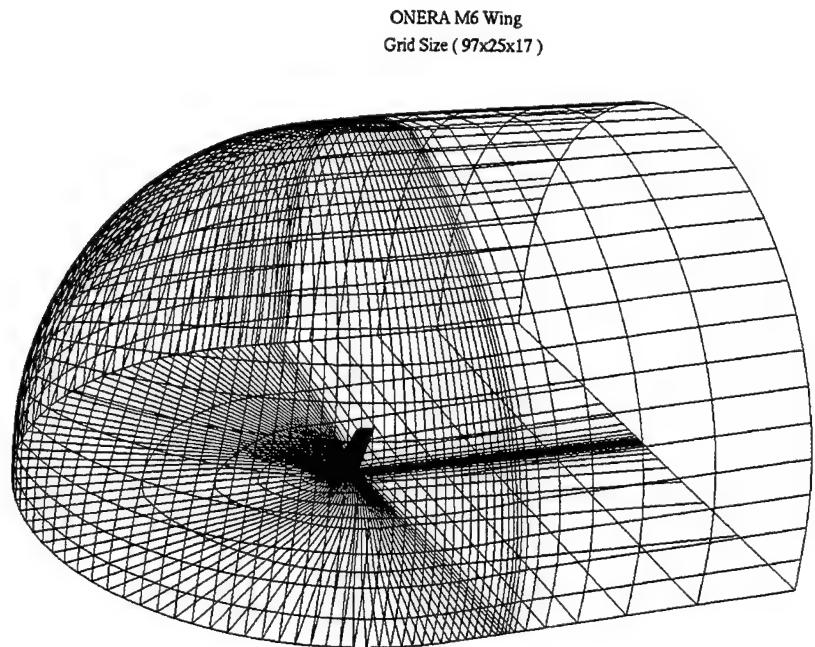


(b) Pressure Distribution

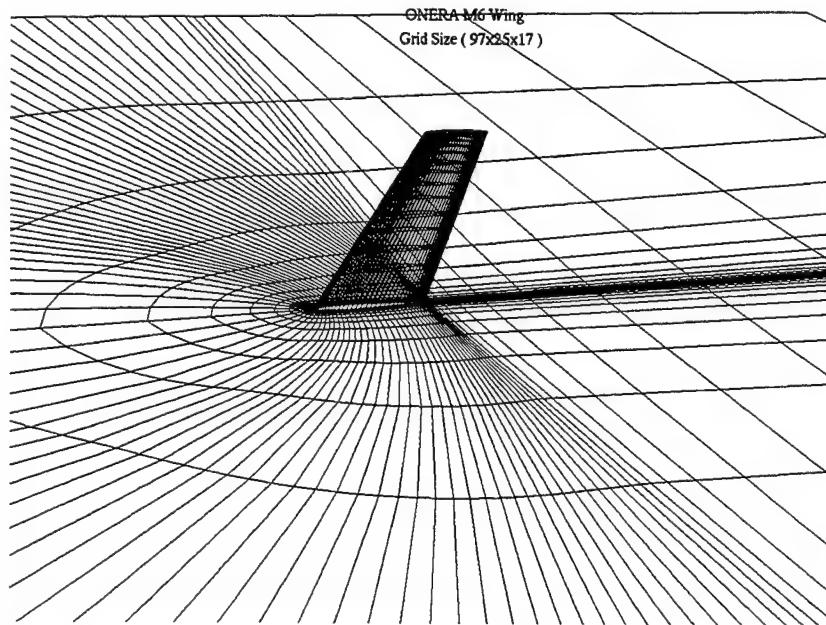


(c) Mach Number Distribution

Figure 6.40 Grid, Pressure Distribution and Mach Number Distribution at  $t=243.5836$

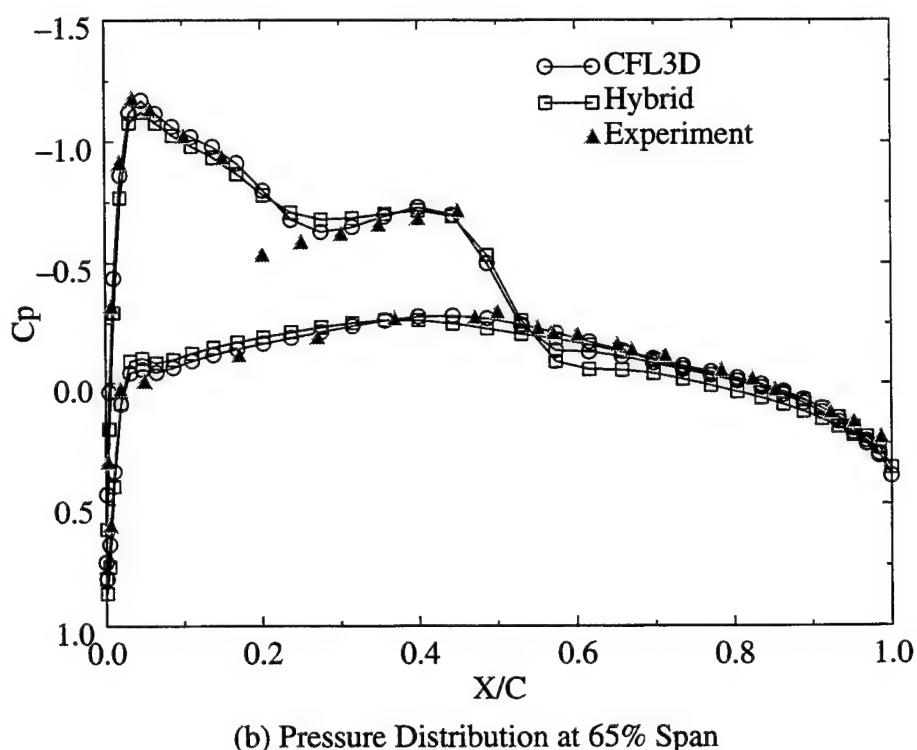
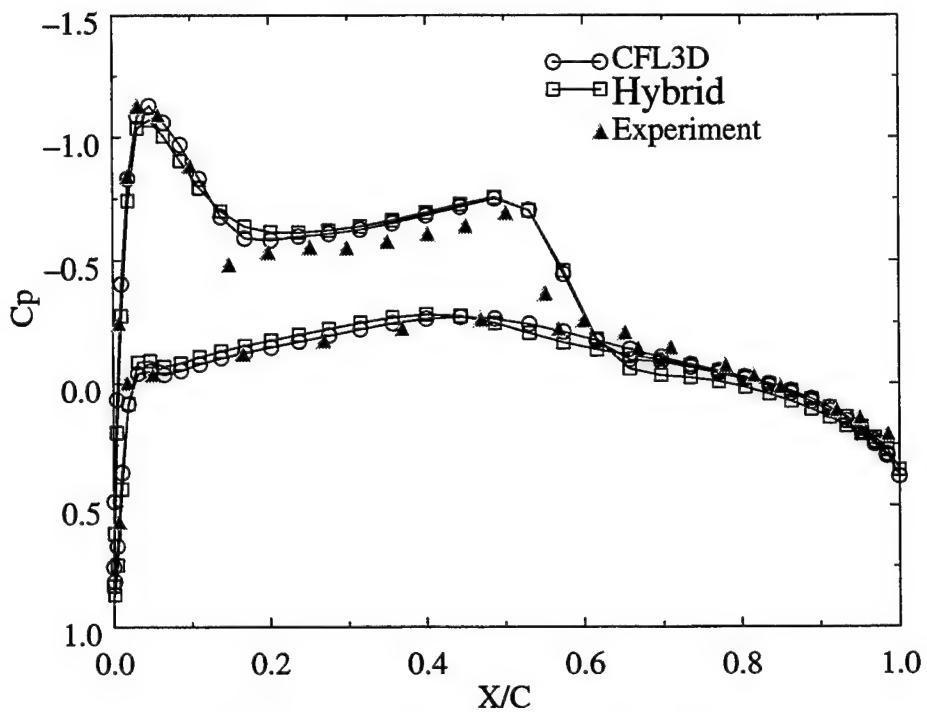


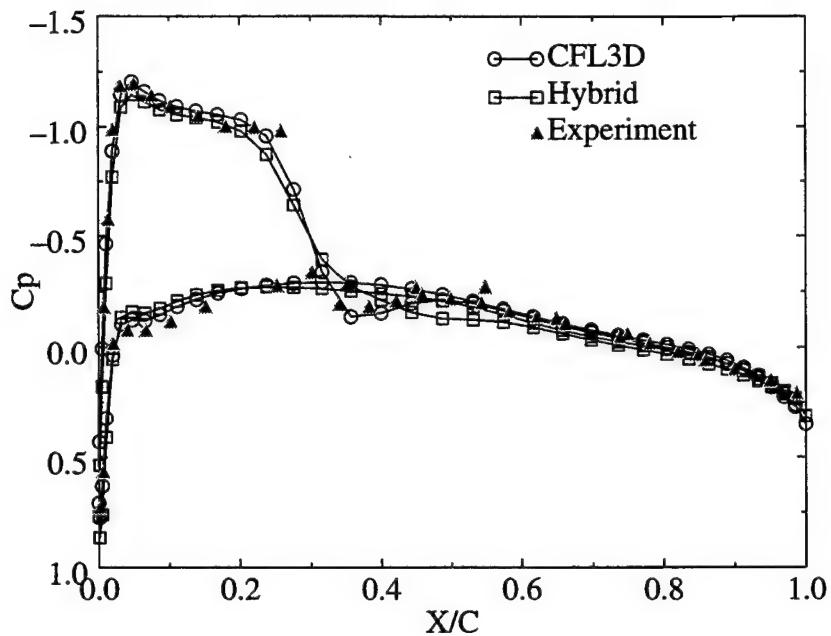
(a) Wing with Outer Boundary Surface



(b) Grid Point Distribution Over and Near the Wing

Figure 6.41 Grid Used for Flow Simulation Over  
ONERA M6 Wing





(c) Pressure Distribution at 90% Span

Figure 6.42 Cp Distribution Over ONERA M6 Wing at Mach Number 0.84 and Angle of Attack 3.06 Degrees

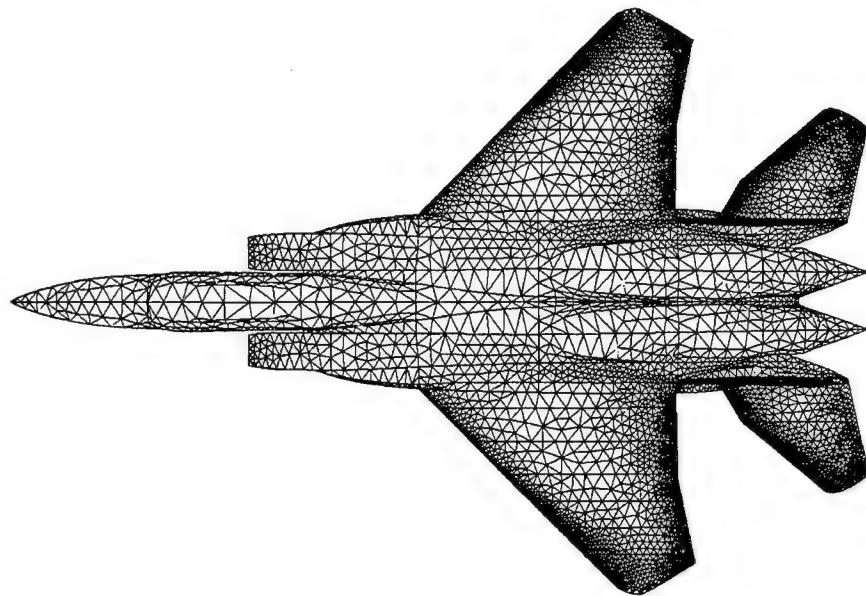


Figure 6.43 Unstructured Surface Grid Used for Flow Simulations Over F-15 Aircraft

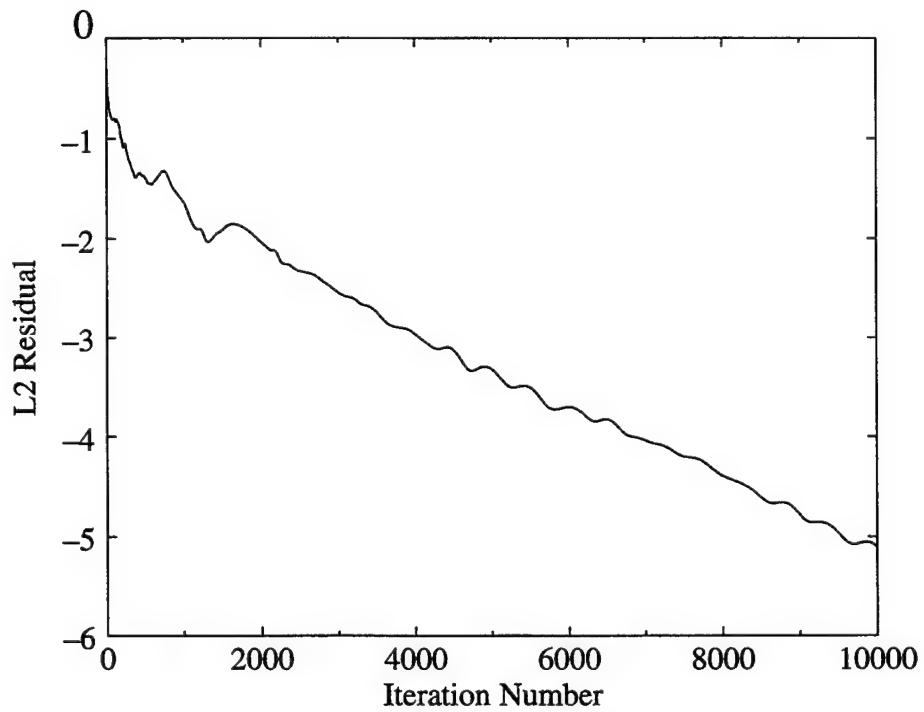


Figure 6.44 Convergence History for F-15 Aircraft Geometry

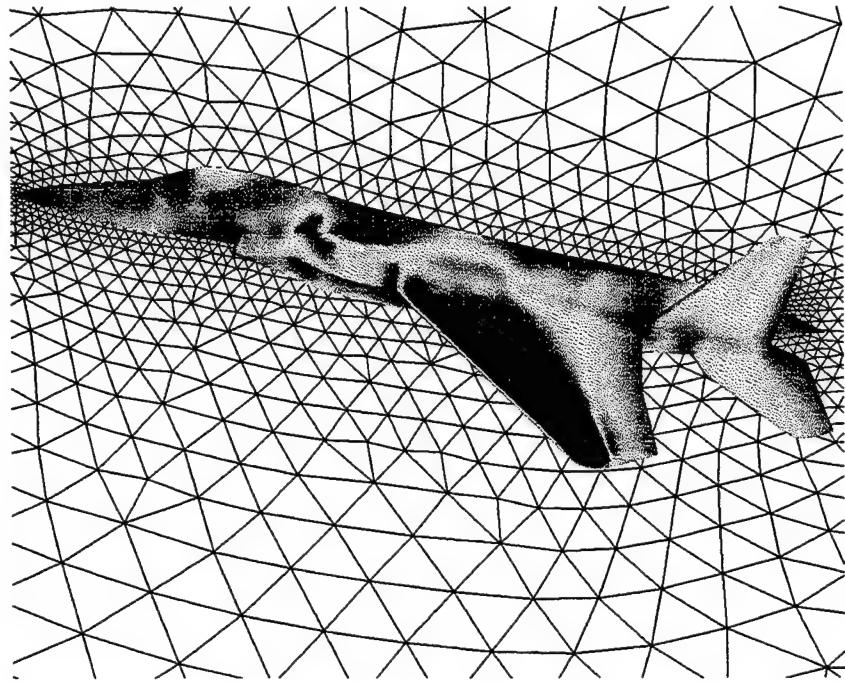


Figure 6.45 Pressure Distribution Over F-15 Aircraft at Mach Number 0.9 and Angle of attack 4.84 Degrees

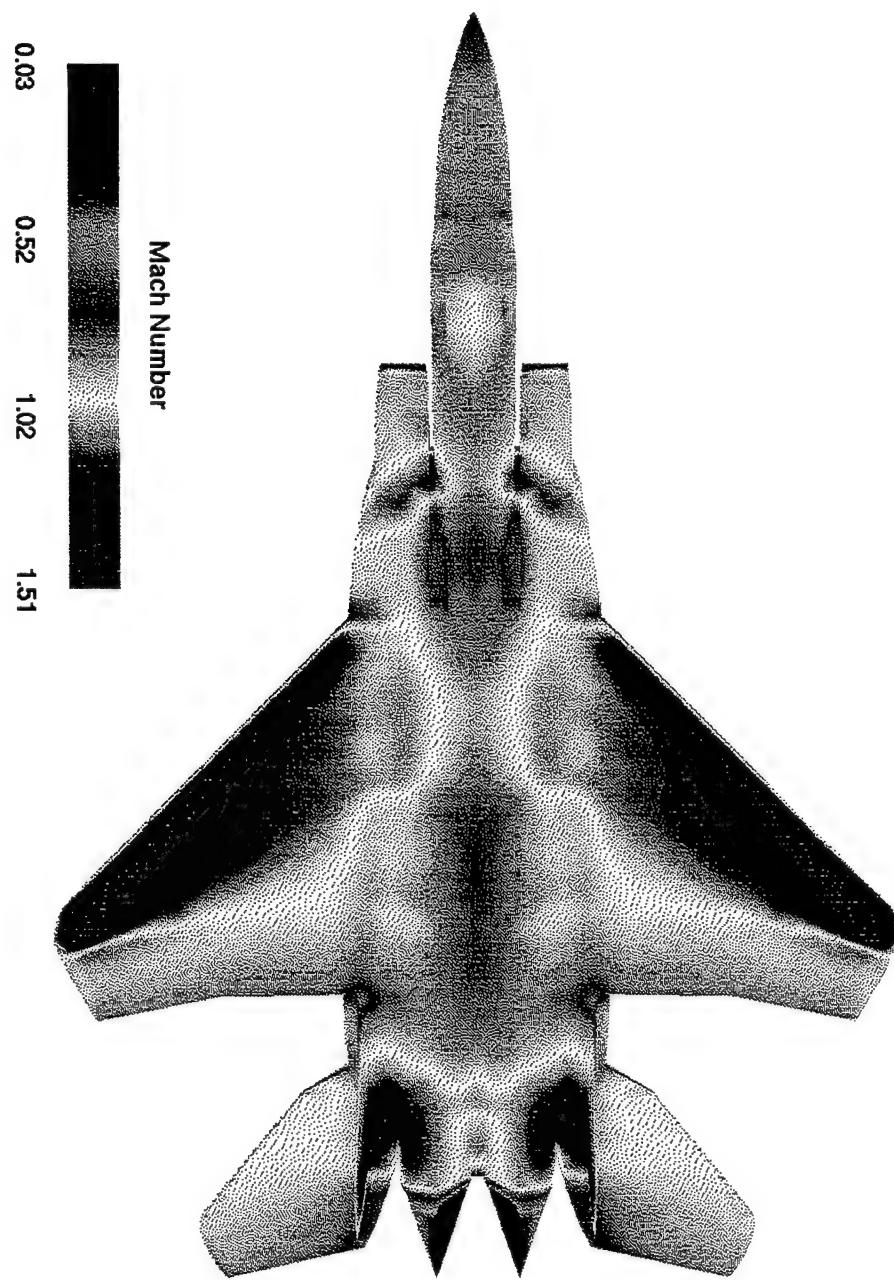
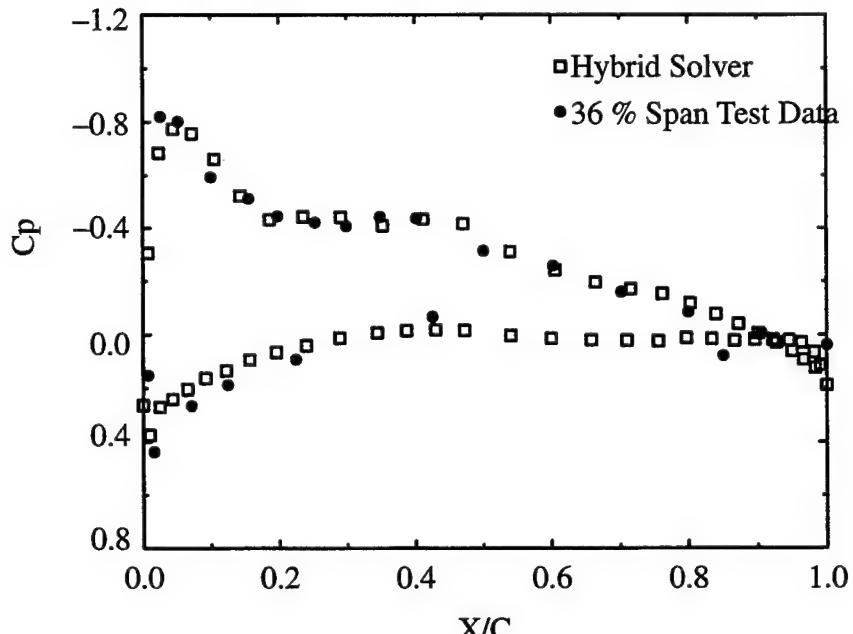
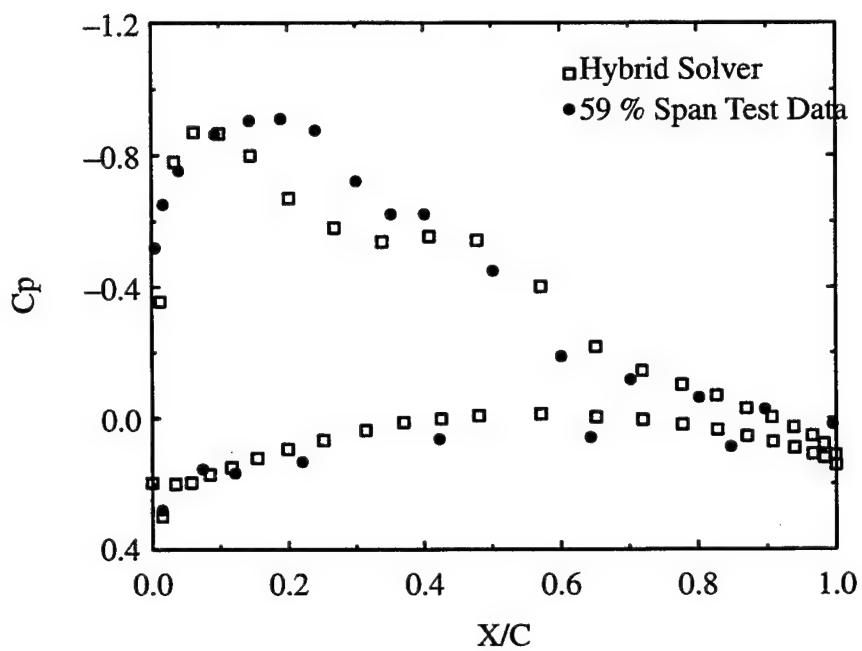


Figure 6.46 Mach Number Distribution Over F-15 Aircraft at Freestream Mach Number 0.9 and Angle of Attack 4.84 Degrees



(a) Pressure Distribution at 36% Span



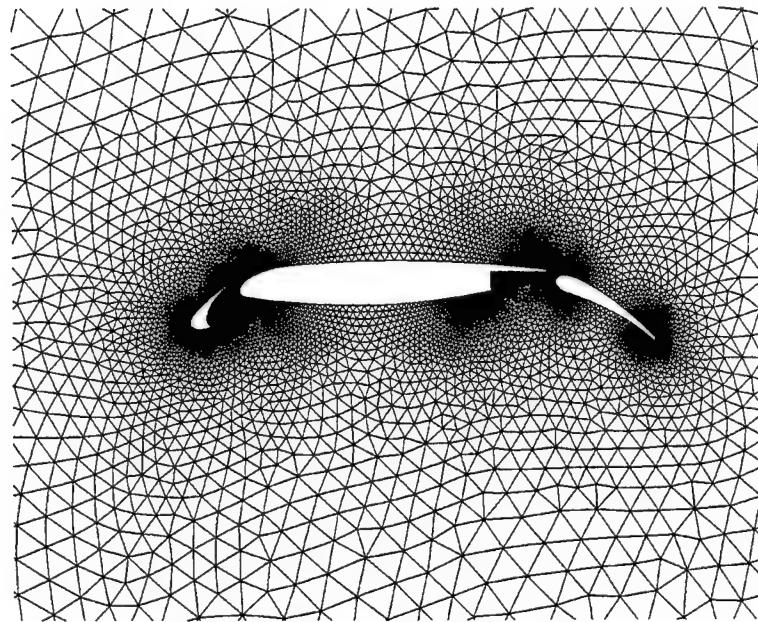
(b) Pressure Distribution at 59% Span

Figure 6.47  $C_p$  Distribution Over the Wing Surface for F-15 Aircraft  
At Mach Number 0.9 and Angle of Attack 4.84 Degrees

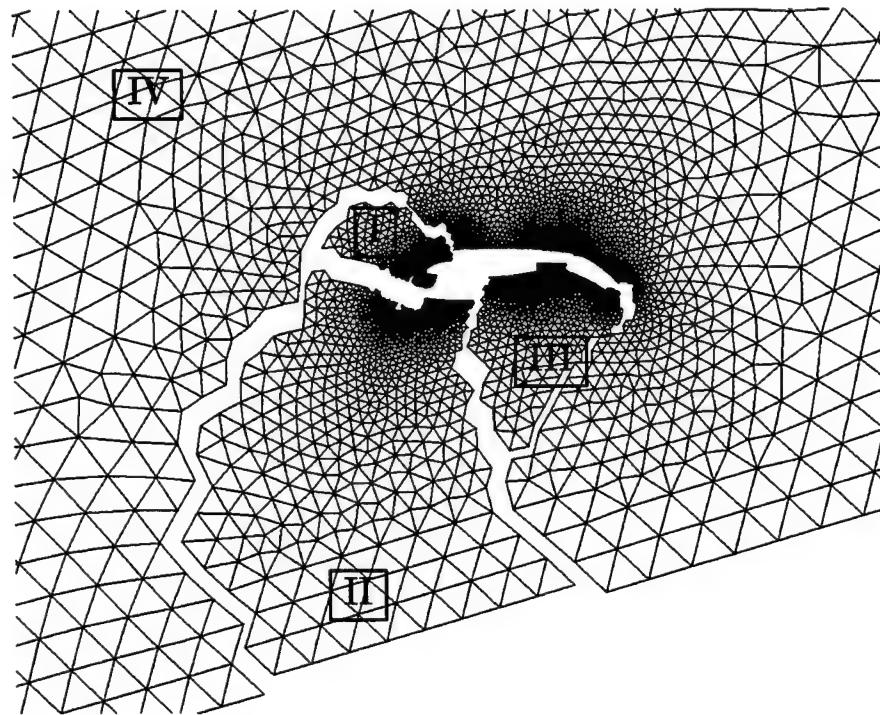
domains (user specified) with minimum number of edge cutting. The details of the graph partitioning algorithms are described in Karypis and Kumar [71].

Different blocks of the partitioned grid are assigned to different processors. The information across the block boundaries are passed as explicit boundary conditions. Message Passing Interface (MPI Gropp. et al. [72]) is used to communicate between the processors. Results are presented for a three element airfoil at a freestream Mach number of 0.2 and an angle of attack of 16.2 degrees. An example of the grid partitioned into four domains is presented in Figure 6.48. The complete grid contains 29830 cells and is divided into four blocks, out of which two blocks contain 7458 cells and the other two contain 7457 cells. The  $C_p$  distribution from the sequential and parallel solver, with four blocks, is compared in Figure 6.49. It can also be seen from Figure 6.49 that the  $C_p$  distribution from the sequential and parallel solvers matches well.

The actual speed up of the parallel solver is compared with the ideal one in Figure 6.50. The SGI processor (ANDY) gives a better speed up as compared to the SUN (superMSPARC) processor, because the data transfer across the processor is faster for the SGI processor.



(a) Grid Around the Three Element Airfoil



(b) Grid Partitioned into Four Blocks

Figure 6.48 Grid for a Three Element Airfoil and its Partition into Four Blocks

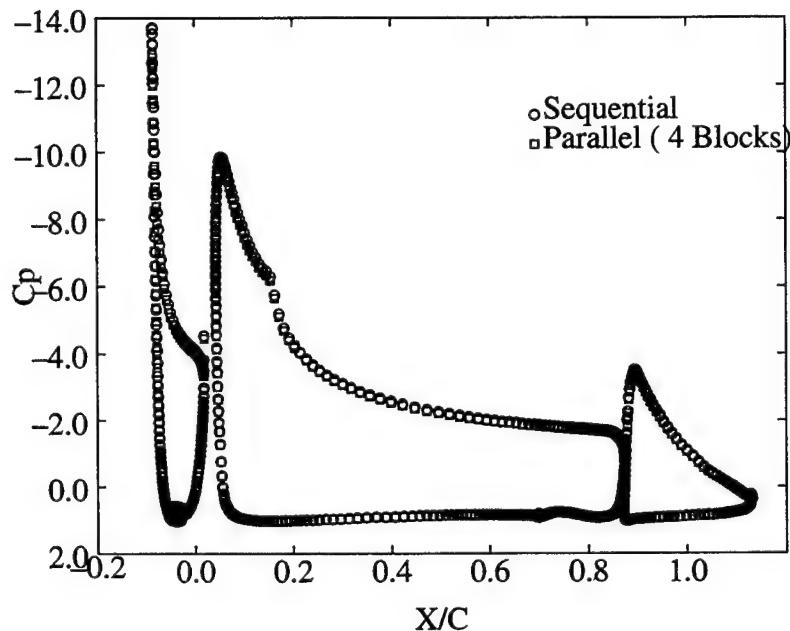


Figure 6.49 Cp Distribution for Sequential and Parallel Solvers for Three Element Airfoil at Freestream Mach Number 0.2 and Angle of Attack 16.2 degrees

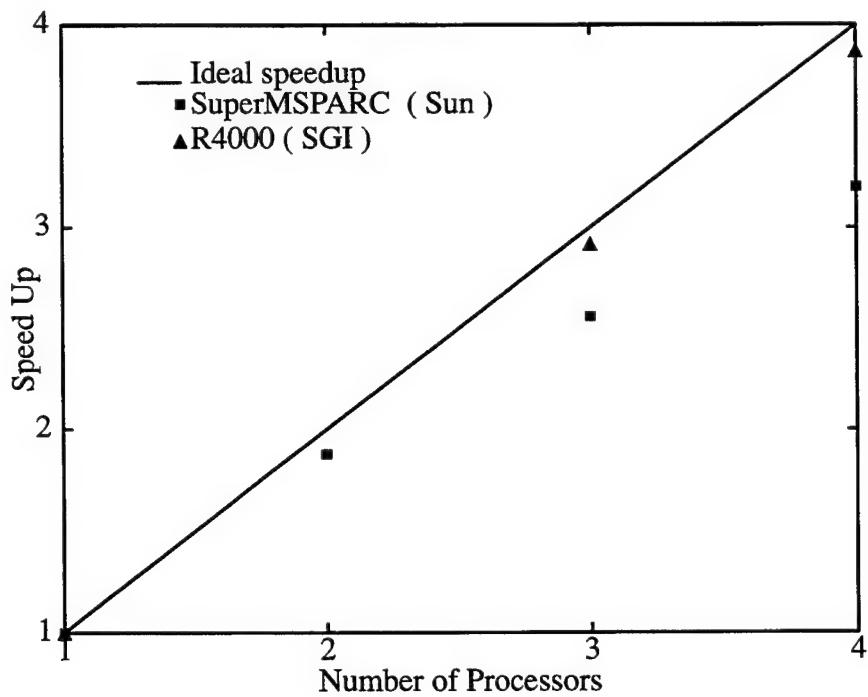


Figure 6.50 Speed Up of Parallel Solver With the Ideal Speed Up

## **7. University–Industry Collaboration**

The broad based objective of the present collaborative project is to provide a forum through which university and industrial researchers can jointly pursue research and development pertinent to hybrid techniques in computational fluid dynamic with a concerted effort in the development of hybrid/generalized grid generation and flow solver technology. This collaboration has worked out very well for both industry and the university. The collaboration has been proven to be very beneficial to participating students. One Ph.D., two masters, and two undergraduate students participated in this program. During the rise of this project, the participating industry management and researchers have gone through various changes (e.g. Boeing merger, leaving of Dr. Rob Rogers from the Teledyne Grown Eng. Corp., changing interests and priorities at McDonnell Douglas Sites, etc.) These changes had an impact on the progress of this project. However, all the targeted objectives of the project were accomplished after additional ten months of no cost extension of this project.

Grid generation and flow simulation capabilities have been transferred to all industrial sites. In particular:

- HYBGD2D, HYBFL2D–3D and MARCH3D systems are now fully operational at the Teledyne Brown Engineering Corporation. The students and faculty have made various trips to Teledyne Brown Engineering to work side-by-side with TBE personnel in validating these systems for TBE applications (Dr. Sonat Praharaj is the new contact person at TBE).
- A completely unstructured grid around complete F15 aircraft was obtained from Boeing, St. Louis (McDonnell Aircraft Co, contact persons: Timothy Gatzke and Raymond Cosner) and the hybrid solutions were compared for validation. The modular of grid generation technology and flow solvers have been transferred to Boeing–St Louis and Boeing, Huntington Beach (McDonnell Douglas Space Systems division, contact persons: Dan Parish and Darrin Fricker).
- The gird generation and hybrid flow simulation capabilities were applied to problems of interest at Lockheed–Martin at Marietta (Contact person: David Vaughna). Various helpful suggestions in this development were supplied by Lockheed Co.
- The simulation capability of the code has been compared with the COBALT system (a CHSSI system, under development at Wright Patterson AFB, contact persons: Don Kinsey and Bill Strang). These results will be reported as a joint publication at a later date.

## 8. References

- [1] Lohner, R., "Adaptive Remeshing for Transient Problems with Moving Bodies," AIAA 88-3737. 1988.
- [2] Marcum, D. L., and Weatherill, N. P., "Aerospace Applications of Solution Adaptive Finite Element Analysis," Computer Aided Geometric Design, vol 12, pp 709-731, 1995.
- [3] Mavriplis, D. J., and Jameson, A., "Multigrid Solution of The Navier-Stokes Equation on Triangular Meshes," AIAA Journal vol 28, No 8, August 1990.
- [4] Mavriplis, D. J., and Venkatakrishnan, V., "Agglomeration Multigrid for the Three Dimensional Euler Equations," AIAA 94-0069. 1994.
- [5] Morgan, K., Peraire, J., and Peiro, J., "The Computation of Three Dimensional Flows Using Unstructured Grids," Computer Methods in Applied Mechanics and Engineering (87), p 335-352, 1991.
- [6] Venkatakrishnan, V., and Mavriplis, D. J., "Computation of Unsteady Flows Over Complex Geometries in Relative Motion," Presented at The First AFOSR Conference on Dynamic Motion CFD. Rutgers University, NJ. June 3-5, 1996.
- [7] Whitaker, D. L., Slack D. C., and Walters, W., "Solution Algorithms for the Two-Dimensional Euler Equations on Unstructured Meshes," AIAA 90-0697. 1990.
- [8] Whitfield, D. L., Janus, J. M., and Simpson, L. B., "Implicit Finite Volume High Resolution Wave Split Scheme for Solving the Unsteady Three Dimensional Euler and Navier-Stokes Equations on Stationary or Dynamic Grids," Engineering and Industrial Research Station Report Number MSSU-EIRS-ASE-88-2. Mississippi State University, MS. 1988.
- [9] Dannenhoffer, J. F., "Automatic Generation of Block Structures – Progress and Challenges," Proceedings of 5th International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields. Mississippi State University, MS. Edited by B. K. Soni, J. F. Thompson, J. Hauser and P. Eiseman, pp 403-412. April 1-5, 1996.
- [10] Shih, M. H., and Soni, B. K. "Geometry Modeling and Multi-Block Grid Generation for Turbomachinery Configuration," Proceedings of the NASA Workshop on Software Systems for Surface Modeling and Grid Generation, Hampton, VA, April 1992.

- [11] Thompson, J. F., "A General Three-Dimensional Elliptic Grid Generation System on a Composite Block Structure," *Computer Methods in Applied Mechanics and Engineering*, vol. 64, p. 377, 1987.
- [12] Kallinderis, Y., Khawaja, A., and McMorris, H., "Hybrid Prismatic/Tetrahedral Grid Generation for Complex Geometries," *AIAA 95-0211*, Presented at 33rd Aerospace Sciences Meeting and Exhibit, Reno, NV. January 9–12, 1995.
- [13] Khawaja, A., McMorris, H., and Kallinderis, Y., "Hybrid Grid for Viscous Flows around Complex 3-D Geometries Including Multiple Bodies," *AIAA 95-1685*. 12th AIAA Computational Fluid Dynamics Conference, June 19–22, 1995, San Diego, CA.
- [14] Lohner, R., and Parikh, P., "Generation of Three Dimensional Unstructured Grids by The Advancing Front Method," *International Journal for Numerical Methods in Fluids*, vol 8, p 1135–1149, 1988.
- [15] Marcum, D. L., and Weatherill, N. P., "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection," *AIAA Journal*, vol 33, no 9, pp 1619–1625. September 1995.
- [16] Mavriplis, D. J., "An Advancing Front Delaunay Triangulation Algorithm Designed for robustness," ICASE Report No 92-49. 1992.
- [17] Mavriplis, D. J., "Unstructured Mesh Generation and Adaptivity," Notes for 26th Computational Fluid Dynamics Lecture Series Program of the von Karman Institute for Fluid Dynamics, Rhode-Saint-Ghene, Belgium, 13–17 March 1995.
- [18] Weatherill, N. P., "Grid Generation by Delaunay Triangulation," Lecture Series in von Karman Institute for Fluid Dynamics 1993–94.
- [19] Baum, J. D., Luo, H., and Lohner, R., "A New ALE Adaptive Unstructured Methodology for the Simulation of Moving Bodies," *AIAA 94-0414*. 32nd Aerospace Sciences Meeting and Exhibit, Reno, NV, January 10–13, 1994.
- [20] Marcum, D. L., and Weatherill, N. P., "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection," *AIAA 94-1926*. 12th AIAA Applied Aerodynamics Conference. Colorado Springs, CO. June 20–22, 1994.
- [21] Cooper, G. K., and Srbough, J. R., "PARC Code: Theory and Usage," AEDC-TR-89-15, Arnold Engineering Development Center, Arnold AFB, 1989.
- [22] Thomas, J., Krist, S., and Anderson, W., "Navier-Stokes Computations of Vortical Flows Over Low-Aspect-Ratio Wings," *AIAA Journal*, vol 28, number 2, pp. 205–212, 1990.
- [23] Marcum, D. L., "Generation of Unstructured Grids for Viscous Flow Applications," *AIAA 95-0212*. 33rd Aerospace Sciences Meeting and Exhibit. Reno, NV. January 9–12, 1995.

- [24] Pirzadeh, S., "Viscous Unstructured Three-Dimensional Grids by the Advancing-Layers Method," AIAA Paper 94-0417, 1994.
- [25] Nakahashi, K., Nozaki, O. Kikuchi, K., and Tamura, A., "Navier-Stokes Computations of Two- and Three-Dimensional Cascade Flows," Journal of Propulsion and Power, vol 5, No 3, pp 320–326, May–June 1989.
- [26] Lohner, R., "Matching Semi-structured and Unstructured Grids for Navier-Stokes Calculations," AIAA 93-3338-CP, Presented at 11th AIAA Computational Fluid Dynamics Conference, Orlando, FL, July 6–9, 1993.
- [27] Sharov, D., and Nakahashi, K., "Hybrid Prismatic/Tetrahedral Grid Generation for Viscous Flow Applications," AIAA Paper 96-2000, 1996.
- [28] Weatherill, N. P., "Mixed Structured-Unstructured Meshes for Aerodynamic Flow Simulation," The Aeronautical Journal, vol 94, No 934, April 1990.
- [29] Kao, K. H., and Liou, M. S., "An Advance in Overset Grid Schemes: From CHIMERA to DRAGON Grids," 2nd Overset Composite Grid and Solution Technology Symposium, October 25–28, Fort Walton Beach, FL, 1994.
- [30] Huang, Chih-Ti., "Hybrid Grid Generation System," Master Thesis, Department of Aerospace Engineering, Mississippi State University. August 1996.
- [31] Mathur, R., Madavan, N. K., and Rajagopalan, R. G., "Solution Adaptive Structured-Unstructured Grid Method for Unsteady Turbomachinery Analysis, Part 1: Methodology & Part 2: Results," Journal of Propulsion and Power, vol 10, No 4, p 576 – 584 & p 585–592
- [32] Tsung, F. L., Loellbach, J., Kwon, O., and Hah, C., "A Three Dimensional Structured/Unstructured Hybrid Navier-Stokes Method for Turbine Blade Rows," AIAA 94-3369. 1994.
- [33] Soetrino, M., Imlay, S. T., and Roberts, D. W., "A Zonal Implicit Procedure for Hybrid Structured- Unstructured Grids," AIAA-94-0645, Presented at 32nd Aerospace Sciences Meeting and Exhibit, Reno, NV. January 10–13, 1994.
- [34] Parthasarathy, V., Kallinderis, Y., and Nakajima, K., "Hybrid Adaptation Method and Directional Viscous Multigrid with Prismatic-Tetrahedral Meshes," AIAA Paper 95-0670. January 1995.
- [35] Spalart. P. R., and Allmaras, S. R., "A One Equation Turbulence Model for Aerodynamic Flows," AIAA 92-0439. 1992.
- [36] Soni, B. K., "Fundamental Numerical Grid Generation", Note For Course ASE 6433 (Unpublished), Mississippi State University, Fall 1994.
- [37] Soni, B. K., "Grid Generation Techniques", Note For Grid Generation Short Course (Unpublished), Mississippi State University December 1994.

- [38] Soni, B. K., "Elliptic Grid Generation System: Control Functions Revised—I", Applied Mathematics and Computation, vol59, pp 151–163, 1993.
- [39] Marcum, D. L., "Unstructured Grid Technology", Class Note for the Course ME 8993 (Unpublished), Mississippi State University, Fall 1995.
- [40] Warsi, Z. U. A., "Fluid Dynamics: Theoretical and Computational Approaches," CRC Press 1993.
- [41] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., "Computational Fluid Mechanics and Heat Transfer," Series in Computational Methods in Mechanics and Thermal Sciences, Hemisphere Publishing Corporation, 1984.
- [42] Barth, T. J., and Jespersen, D. C., "The Design and Application of Upwind Schemes on Unstructured meshes," AIAA 89-0366. 1989.
- [43] Whitaker, D. L., "Three Dimensional Unstructured Grid Euler Computations Using A Fully Implicit Upwind Method," AIAA 93-3337-CP. 1993.
- [44] Jameson, A., Baker, T. J., and Weatherill, N. P., "Calculation of Inviscid Transonic Flow Over a Complete Aircraft," AIAA 86-0103. 1986.
- [45] Roe, P. L., "Approximate Riemann Solvers, Parameter Vector, and Difference Schemes," Journal of Computational Physics, vol. 43, pp 357–372, 1981.
- [46] LeVeque, R. J., "Numerical Methods for Conservation Laws," Lectures in Mathematics. Second Edition, Birkhauser Verlag. 1992.
- [47] Quirk, J. J., "A Contribution to the Great Riemann Solver Debate," ICASE Report No. 92-64, NASA Contractor Report 191409. November 1992.
- [48] Frink, N. T., Parikh, P., and Pirzadeh, S., "Aerodynamic Analysis of Complex Configurations Using Unstructured Grids," AIAA 91-3292. 1991.
- [49] Cabello, J., Morgan, K., and Lohner, R., "A Comparison of Higher Order Schemes Used in a Finite Volume Solver For Unstructured Grids," AIAA 94-2293. Presented at the 25th AIAA Plasmadynamics and Lasers Conference, Colorado Springs, CO. June 20–23, 1994.
- [50] Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," AIAA 93-0880. January 1993.
- [51] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time Stepping Schemes," AIAA Paper 81-1259, 1981.
- [52] Koomullil, R. P., Soni, B. K., and Huang, Chih-Ti., "Navier–Stokes Simulation on Hybrid Grids," AIAA 96-0768. Presented at 34th Aerospace Sciences Meeting and Exhibit, Reno, NV. January 15–18, 1996.

- [53] Turkel, E., and Van Leer, B., "Flux–Vector Splitting and Runge–Kutta Methods for the Euler Equations," ICASE Report Number 84–27, NASA CR 172415, June 1984.
- [54] Whitfield, D. L., and Taylor, L., "Discretized Newton–Relaxation Solution of High Resolution Flux–Difference Split Schemes," AIAA 91–1539, June 1991.
- [55] Whitfield, D. L., "An Implicit Solution of Roe's Approximate Riemann Solver Using True and Approximate Jacobians, Including Discretized and Sparse Matrix Update," MSSU–EIRS–ERC–93–1, October 1992.
- [56] Whitfield, D. L., "Newton Relaxation Schemes for Nonlinear Hyperbolic Systems," MSSU–EIRS–ASE–90–3, Mississippi State University, MS. 1990.
- [57] Vanden, K. J., "Direct and Iterative Algorithms for the Three–Dimensional Euler Equations," Dissertation, Department of Aerospace Engineering, Mississippi State University, December 1992.
- [58] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Non–symmetric Linear Systems," SIAM Journal of Sci. Stat. Comp. 7, pp 865–869. 1986.
- [59] Hamann, Bernd., Private Communications, 1995.
- [60] Whitfield, D. L., and Janus, J. M., "Three–Dimensional Unsteady Euler Equations Solution Using Flux Vector Splitting," AIAA 84–1552, June 1984.
- [61] Allmaras, S. R., Private Communications, 1996.
- [62] Thomas, P. D., and Lombard, C. K., "Geometric Conservation Law and Its Applications to Flow Computations on Moving Grids," AIAA Journal, vol 17, number 10, pp 1030–1037, October 1979.
- [63] Janus, M. J., "Advanced 3–D Algorithm for Turbomachinery," Dissertation, Department of Aerospace Engineering, Mississippi State University, May 1989.
- [64] Singh, K. P., Newman, J. C., and Baysal, O., "Dynamic Unstructured Method for Flows Past Multiple Objects in Relative Motion," AIAA Journal, vol. 33, number 4, pp 641–649, April 1995.
- [65] Koomullil, R. P., Soni, B. K., and Huang, Chih-Ti., "Unsteady Flow Simulations on Hybrid Grids," Presented at The First AFOSR Conference on Dynamic Motion CFD. Rutgers University, NJ. June 3–5, 1996.
- [66] Fletcher, D. Q., "Mechanics of Materials," CBS College Publishing. 1985.
- [67] Baldwin, B. S., and Barth, T. J., "A One–Equation Turbulence Transport Model for High Reynolds Number Wall–Bounded Flows," NASA Technical Memorandum 102847. August 1990.

---

- [68] Grimes, D. B., "Efficient Grid Generation Techniques for Producing Two-Dimensional, Solution-Adapted Unstructured Grids," Master Thesis, Department of Mechanical Engineering, Mississippi State University. August 1994.
- [69] Landon, R. H., "NACA0012 Oscillating and Transient Pitching," Data Set 3 in AGRAD-R-702, Compendium of Unsteady Aerodynamic Measurements. August 1982.
- [70] Michal, T., and Halt, D., "Development and Application of an Unstructured Grid Flow Solver for Complex Fighter Aircraft Configurations," AIAA 95-1785. Presented at the AIAA 13th Applied Aerodynamics Conference, San Diego, CA. June 19–22, 1995.
- [71] Karypis, G., and Kumar, V., "Metis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0," Department of Computer Science, University of Minnesota, August , 1995.
- [72] Gropp, W., Lusk E., and Skjellum, A., "Using MPI, Portable Parallel Programming with the Message-Passing Interface", Scientific and Engineering Computation Series, 1994.
- [73] Koomullil, R. P., Soni, B. K., and Huang, Chih-Ti., "Flow Simulations on Generalized Grids," Proceedings of 5th International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields. Mississippi State University, MS. Edited by B. K. Soni, J. F. Thompson, J. Hauser and P. Eiseman, pp 527–536. April 1–5, 1996.
- [74] Bowyer, A., "Computing Dirischlet Tessellations" The Computer Journal, vol 24, No 2, p 162 – 166, 1981.
- [75] Janus, M. J., "The Development of a Three-Dimensional Split Flux Vector Euler Solver with Dynamic Grid Applications," Master Thesis, Department of Aerospace Engineering, Mississippi State University. August 1984.